NASA CR-73107

# ANALYSIS AND DESIGN OF SUPERSONIC WING-BODY COMBINATIONS, INCLUDING FLOW PROPERTIES IN THE NEAR FIELD

## PART II—DIGITAL COMPUTER PROGRAM DESCRIPTION

By E. LaRowe and J. E. Love

D6-15044-2

August 1967



Prepared under Contract No. NA 2-3719 by

The Boeing Company
Commercial Airplane Division
P.O. Box 707
Renton, Washington

for
AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

CONTENTS

# 1. INTRODUCTION

This part of the document describes the digital computer program developed to analyze and design wing-body combinations at supersonic speeds. The program is an extension of an earlier program that is presented in references 1 and 2. The theoretical considerations underlying the method used in the program are discussed in reference 3.

The complete program consists of five sections: Geometry Definition, Geometry Transformation, Geometry Paneling, Aerodynamics, and Flow Visualization. The first three sections present, in a form acceptable to the Aerodynamics section, a geometric description of the configuration under study. A wing-alone, body-alone, or wing-body combination configuration may be defined. The Geometry sections redefine the body as an equivalent body of circular cross section. The wing and that part of the body in the region aft of the wing leading-edge intersection are subdivided into a large number of panels. The wing may be defined to have thickness or may consist of a camber surface only. The Aerodynamics section then solves the problem by first computing the velocity components induced by the various singularities that represent the configuration. The body thickness and camber effects are represented by line sources and doublets along the body axis, and wing thickness effects are represented by a distribution of sources located in the wing reference plane. The effects of camber and incidence of the wing and the effects of any residual interferences are simulated by a surface distribution of vortices on the wing and body. When the velocity components are computed, the program determines the strengths of the singularities required to satisfy the given boundary conditions. In addition to solving the problem of optimizing wing camber shape, the program will also solve the direct analysis problem of calculating pressures, forces, and moments on a given configuration, and the indirect design problem of determining the wing shape for a given lift distribution. The program selects the boundary conditions appropriate to the particular optimization, direct or indirect problem being considered, and solves the problem. The last section of the program, Flow Visualization, computes velocities at selected field points and integrates streamlines about the configuration.

This part of the document is a guide for program maintenance. An outline of the program structure is followed by a detailed description of each section of the program. The section on program usage contains the input format, timing and output estimates, and the required machine components. Input and output

formats of a sample case are also included. The section on program records contains flow charts, overlay diagrams, and descriptions of each subroutine. A program listing and other appendixes complete this part of the document.

## 2. DESCRIPTION

### 2.1 Geometry Definition

This part of the program defines geometrically a wing, a body, or a wing-body combination. The definition, in the form of body meridian lines and wing percent chord lines, is written on a tape that is read by the Geometry Transformation section of the program

Meridian lines may be visualized as stringers running lengthwise along the surface of the body. Each meridian line consists of a series of straight-line segments connecting points on the body surface; adjacent points are connected by straight lines. Constant percent chord lines, much like meridian lines, run along the wing surface in a spanwise direction.

A body surface is specified by a set of defining sections normal to the X-axis. Points on defining sections are given in polar coordinate form ( $\rho$ , $\theta$ ) or in rectangular coordinates (Y, Z). The number of points in each section must be the same as the number of meridian lines. Since the X-coordinate of each defining section is known, the X, Y, Z-coordinates of each point in a section can be computed. The first meridian line is constructed by joining the first point in each section; succeeding meridian lines are similarly constructed. As an option, the meridian lines may be "enriched"; additional meridian line points are inserted by three-dimensional interpolation from a space curve through the original points.

Both upper and lower wing surfaces may be defined. Only the construction of upper wing percent chord lines will be discussed, because later sections of the program are presently limited to this case. Although more complicated, the procedure is similar to that used to define the body. Percent chord lines (enriched if desired) on the surface of the wing are computed from a given set of control chords. These control chords must be parallel to the X-Y plane but are not necessarily parallel to the X-Z plane. In order to locate each control chord in space, sufficient data are given to find the intersections of the chord with the leading and trailing edges of the given planform. The distance between these intersections is called (in this discussion only) the "oblique chord length," because control chords need not be streamwise.

The projection of wing planform in the X-Y plane is specified by a set of points on the leading and trailing edges; the number of points in each set need not be the same. The planform may extend past the control chord region, but

the program will reject any chord which does not intersect both the leading and trailing edges. The planform may be open or closed at the wing tip. A planform is closed if the leading edge and trailing edge intersect at the wing tip; the generated percent chord lines will then converge to that point. If the leading and trailing edges do not intersect the planform is open.

A control chord is given as a set of points in a two-dimensional coordinate system. These points are scaled (except for zero-length chords) so that the difference in abscissa of the first and last points is equal to the oblique chord length. This scaled chord is then placed on the planform. Wing percent chord lines are constructed between scaled control chords.

The wing-body intersection option calculates the intersection of each wing percent chord line with the body surface. The wing percent lines are then truncated at the body surface.

All variable-length arrays are stored in a single buffer of 10,000 cells. As such there are no individual limits on the number of body stations, number of meridian or percent lines, number of chords or number of points defining the leading and trailing edges of the wing. (However, the data are scanned and excessively large arrays are rejected.) Maximum sizes for data arrays are usually set by later program segments.

## 2.2 Geometry Transformation

This section of the program reads the body and wing definitions from a tape written by the previous section. It then transforms the body and wing to a new coordinate system, orients the wing planform parallel to the x-y plane in this transformed system, and finds the intersections of wing percent chord lines with the body surface. Results are written on tapes for use by later program sections.

Three types of configurations may be handled: body alone, wing alone, or wing-body combination. If a body surface is involved only half of it is given; it is assumed to be symmetrical about the X-Z plane. Then a new x, y, z coordinate system is set up with the origin at the centroid of the forward end of the body and the x-axis passing through the aft end centroid; the z-axis lies in the X-Z plane. In the new coordinate system the body ends (unless of zero radius) are not necessarily parallel to the y-z plane; this is corrected by a

linear adjustment of all x-coordinates of the body points. Also, the new body length (distance between body end centroids) may differ from the original length. The original set of body defining stations is converted to the new coordinate system by a transformation of the form $x = \alpha X + \beta$ where $\alpha$ and $\beta$ are constants. The body radius and centroid are computed at each of these stations. In a body alone case, the radius and centroid are then found by interpolation at a given number of equally spaced stations.

In a wing-body combination case, the wing is also transformed to the x, y, z coordinate system. Then the intersections of the wing leading and trailing edges with the body surface are computed; let $z_A$ be the average of the z-coordinates of these two points. The wing is effectively flattened by changing each z-coordinate of the wing to $z_A$. The body is then intersected by the wing percent chord lines of the flattened wing.

The calculation of body radius and centroid is now resumed. A set of equally spaced stations is chosen between the forward end of the body and the station at which the wing leading edge intersects the body; a similar set is selected from there on aft. The calculation of the interval between stations in each region is subject to the constraints that the total number of stations must agree with a given number and that the two intervals must be approximately the same in each region.

In a wing alone case, the transformation is omitted. However, the wing is flattened; $z_A$ is taken as the average of the z-coordinates of the inboard point on the leading and trailing edge percent chord lines. Bodies defined by a maximum of 16 meridan lines of 90 points each and wings of 16 percent chord lines of 90 points each can be processed.

## 2.3 Geometry Paneling

This program section panels the body and/or wing and provides additional data required by the following section (Aerodynamics). Two cases can be given as input to the Geometry Paneling section under the present program Overlay setup: the wing-body case and the wing alone case. The body alone case does not use the paneling section.

For a wing-body case, the body definition that is written on a scratch tape by the previous section (Geometry Transformation) is read into storage by an

input subroutine, INPUTB. The paneling is accomplished by combining this definition with data for the body cutting planes, read from input cards. A body cutting plane is oriented normal to the X-axis. The body panel areas are computed, and the panel centroid and control point coordinates are calculated. Streamwise chords are constructed through the panel control points, and the chord lengths are calculated. Finally the panel $\theta$ -inclination and $\alpha$-incidence angles are computed. The paneling description and the additional geometric data are written on an output tape for printout purposes and on a scratch tape for use in the Aerodynamics section.

The wing paneling that is required for either the wing-body or the wing alone case is obtained similarly. The wing definition written by the Geometry Transformation section is read into storage arrays by an input subroutine, INPUTW, and used with card input data to provide a wing paneling scheme. Again the wing panel areas, chord lengths, and centroid and control point coordinates are calculated. The inclination and incidence angles are defined to be zero for wing panels since the wing is assumed to be a flat plate with no dihedral at zero incidence. As an option, camber and thickness ordinates and slopes are computed from airfoil ordinate tables. The wing paneling data are written on output and scratch tapes.

Certain program restrictions are present. Though a body that closes to a point can be defined in the definition section, such a body can not be paneled. For this case the leading row of panels would consist of triangular, not quadrilateral panels. This difficulty is avoided by excluding body cutting planes that intersect the body at the nose.

A second restriction requires a flat wing for all program cases. Under the current Overlay scheme, however, the Geometry Transformation section always provides such a definition. Hence this restriction results in no difficulty to the user.

When errors occur, an error message appears, execution of the present case is terminated, a partial data printout is given, and the next case is processed.

## 2.4 Aerodynamics

The aerodynamics problem is solved in three stages: (1) geometry input and calculation of the velocity components, and (2) a preprocessing stage, which is preparation for (3) the calculation of the aerodynamic cases. Control of the flow through the main links of this section of the program is provided by the subroutine AERO and is dependent on the configuration (wing alone, body alone, or wing-body combination) used.

The wing is defined by a number of panels. For an isolated body, the definition is given as an equivalent body of circular cross-section. In order to compute the interference effects on the body, for wing-body combinations, the body surface aft of the wing leading-edge intersection is also represented by panels. The geometric definition of the configuration is read from tapes generated by the geometry sections; this information also includes the areas, centroids, and control points of each panel.

Velocity components due to the various singularities representing the configuration are computed and stored on tape. For a wing defined to have thickness, the program computes velocity components induced by sources located in the reference plane of the wing. The effects of body thickness, camber, and incidence are simulated by placing line sources and doublets along the axis of the body. The effects of camber and incidence of the wing, and any residual interference effects, are represented by a surface distribution of vorticity on the wing and body panels. An option is provided to print out velocity components due to the individual singularities.

The preprocessing stage consists of the following computations. The velocity components are reordered and rewritten on tape. Several matrix operations are performed on the matrix of the aerodynamic influence coefficients (the normal components of velocity due to the vortex distributions), and the results of these are also written on tape. In particular, if the matrix is represented by the following sub-matrices:

$$\begin{bmatrix} [A_{BB}] & [A_{BW}] \\ [A_{WB}] & [A_{WW}] \end{bmatrix}$$

where $[A_{BB}]$ is the influence on the body due to the body

$[A_{WB}]$ is the influence on the wing due to the body

$[A_{BW}]$ is the influence on the body due to the wing

$[A_{WW}]$ is the influence on the wing due to the wing

Then the following matrices are formed:

$$[D] = [A_{WB}] \cdot [A_{BB}]^{-1}$$

$$[E] = [A_{BB}]^{-1} \cdot [A_{BW}]$$

$$[A_R] = [A_{WW}] - \left[[A_{WB}] \cdot [A_{BB}]^{-1} \cdot [A_{BW}]\right]$$

$A_R$ is referred to as the "reduced" matrix of the aerodynamic influence coefficients.

For optimization of the wing camber surface shape, the Lagrange multiplier technique is applied, and the coefficients of the resulting system of equations are computed (see equation (153), Part I):

$$\begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & & & -A_2 & \cdot \\ \vdots & & & \vdots & \vdots \\ \cdot & & & -A_N & -x_N A_N \\ -A_1 & \cdot & -A_N & 0 & 0 \\ -x_1 A_1 & \cdot & -x_1 A_N & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \bar{L} \\ \bar{M} \end{Bmatrix}$$

This operation results in the formation of two matrices. The first matrix includes wing lift and pitching moment constraints, and the second only the wing lift constraint. For the latter case, the row and column of the matrix corresponding to $\lambda_2$ is omitted. Both matrices are inverted and stored on tape.

The aerodynamic cases can now be computed. Additional input cards are used to request the following cases:

1) To compute the wing camber and twist, given the pressure distribution of the wing
2) To compute the pressure distribution, given the geometry of the configuration
3) To optimize the wing shape for minimum drag, given the wing lift or wing lift and pitching moment constraints

In each case a drag polar may be computed by inputing a series of incremental angles of attack. If a wing with thickness is specified in the input geometry, the program provides an option to compute the pressures due to thickness and to add them to the pressures due to camber. Another option selects either the linear, nonlinear, or exact isentropic equation for computing pressure coefficients. For a given configuration and Mach number, any number of aerodynamic cases may be computed.

Upon completion, the Aerodynamics section either transfers control to the Flow Visualization section if this additional analysis is requested or returns control to the main link.

## 2.5 Flow Visualization

This section of the program provides a method of analyzing the flow about a wing, body, or wing-body combination. The configuration is first processed by the Geometric and Aerodynamic sections of the program. The necessary information is transmitted to subroutine FLOVIZ via an intermediate tape (see Appendix D, Tape Formats). This tape may be saved for future runs, thereby eliminating the necessity of rerunning through the Geometric and Aerodynamic sections.

The method of aerodynamic influence coefficients is also applied to the calculation of field velocity components. The perturbation velocities in the field are determined by summing the product of the elementary influence components and their appropriate singularity strengths. From the perturbation velocities, the local flow vectors and pressure coefficients in the field are computed.

Three methods are available for presenting the flow-field data. First, streamlines may be generated by computing the velocity at discrete points and

integrating by a variable-step Adams-Moulton predictor-corrector method. Second, the flow may be examined at points of a two-dimensional or three-dimensional rectangular grid by specifying the increments of spacing along each axis and an origin for the grid. The two-dimensional grid may be skewed by specifying an implicit axis and a vector formed by the grid origin and another point. Finally, the flow may be analyzed at individual field points.

A special printing option provides the capability of examining the velocity and pressure contributions of the various singularities individually. The results of this option, if requested, immediately follow the total velocity components and pressure coefficient, and are labelled as to type of singularity.

The flow through the Flow Visualization section of the program is controlled by subroutine FLOVIZ. Upon completion of all requested options, control is returned to the subroutine AERO.

# 3. USE OF THE PROGRAM

## 3.1 Machine Components

The program is coded in FORTRAN IV and MAP for the IBM 7090/7094 (32K) digital computer under the control of the Systems Monitor, IBSYS Version 13. As the program is organized to utilize the Overlay feature of the Loader, IBLDR, one of the system units must be used as the input-output tape on which the links are written. This unit is specified on the $ORIGIN control card according to the procedure outlined in reference 2.

In addition to the input and output tapes, the program uses seven tape units for scratch purposes. The choice of tape units to be used will depend on the particular computer installation, and tapes must be changed as required. Subroutine OPCAMI (see section 4.2) initializes all the tape units by assigning a logical number to each. To make the necessary changes, it will be sufficient to change only the logical designations in this subroutine.

## 3.2 Program Timing and Output Estimates

The computer time and number of lines of printout for a single configuration can be estimated from the following equations based on experience on the IBM 7094/M2.

$$\text{Time (minutes)} = 2.5 + .5G\,(4. \times 10^{-4} \times P^2) \times A + C + .1F$$

where G indicates type of paneling:

= 0., no paneling

= 1., wing paneling only

= 2., wing and body paneling

P is number of panels. If no paneling is required, use P = 10.

A indicates aerodynamic calculations:

= 0., no aerodynamic calculations

= 1., aerodynamic calculations requested

C is number of aerodynamic cases. Each of the following is considered a case:

Wing optimization case;

Direct aerodynamic case;

Indirect aerodynamic case;

Each angle of a polar series.

F is total number of field points, grid points, and streamline points required for flow visualization calculations.

$$\text{Output (lines)} = 100 + \left[500 + (10 \times P^{1/2}) \times V \times C\right] \times T + F$$

where T indicates type of case:

= 1., wing-alone case

= 2., body-alone case

= 3., wing-body case

V indicates velocity component printout:

= 1., no velocity component printout:

= 2., velocity components requested

## 3.3 Input Data Format

SUMMARY

The following chart summarizes the input data cards required to analyze each of the three basic configurations.

| Geometry Definition Card Set | Wing-Alone | Body-Alone | Wing-Body |
|---|---|---|---|
| Card 1D | Yes | Yes | Yes |
| Cards 2D-9D | No | Yes | Yes |
| Cards 10D-18D | Yes | No | Yes |
| Cards 19D-20D | No | No | Yes |
| Card 21D | Opt | Opt | Opt |
| Card 22D | Yes | Yes | Yes |

| Geometry Paneling Card Set | Wing-Alone | Body-Alone | Wing-Body |
|---|---|---|---|
| Cards 1P-2P | Yes | Yes | Yes |
| Card 3P | Yes | No | Yes |
| Cards 4P-7P | No | No | Yes |
| Cards 8P-13P | Yes | No | Yes |
| Card 14P | Yes | No | Yes |

| Aerodynamic Card Set | Wing-Alone | Body-Alone | Wing-Body |
|---|---|---|---|
| Cards 1A-6A | Yes | Yes | Yes |
| Cards 7A, 8A-8AA, 9A-9AA | No | Yes | Yes |
| Cards 10A-10AA (or -10AB) | Opt (if CASE = 1.) | No | Opt |
| Cards 11A-11AA, 12A-12AA (or -12AC) | Opt (if CASE = 2.) | No | Opt |
| Card 13A | Opt (if CASE = 3.) | No | Opt |
| Cards 14A-14AA (or -14AB) | Opt (if THICK = 1.) | No | Opt |
| Cards 15A | Opt (if POLAR > 0.) | No | Opt |
| Card 16A | Yes | Yes | Yes |

| Flow Visualization Card Set | Wing-Alone | Body-Alone | Wing-Body |
|---|---|---|---|
| Cards 1F-2F | Yes | Yes | Yes |
| Cards 3F, 3FA, 3FB, 3FC | Opt (GRIDS option) | Opt | Opt |
| Cards 4F, 4FA, 4FB, 4FC | Opt (STREAMLINES option) | Opt | Opt |
| Cards 5F, 5FA | Opt (POINTS option) | Opt | Opt |
| Card 6F | Yes | Yes | Yes |
| Terminal Card | Yes | Yes | Yes |

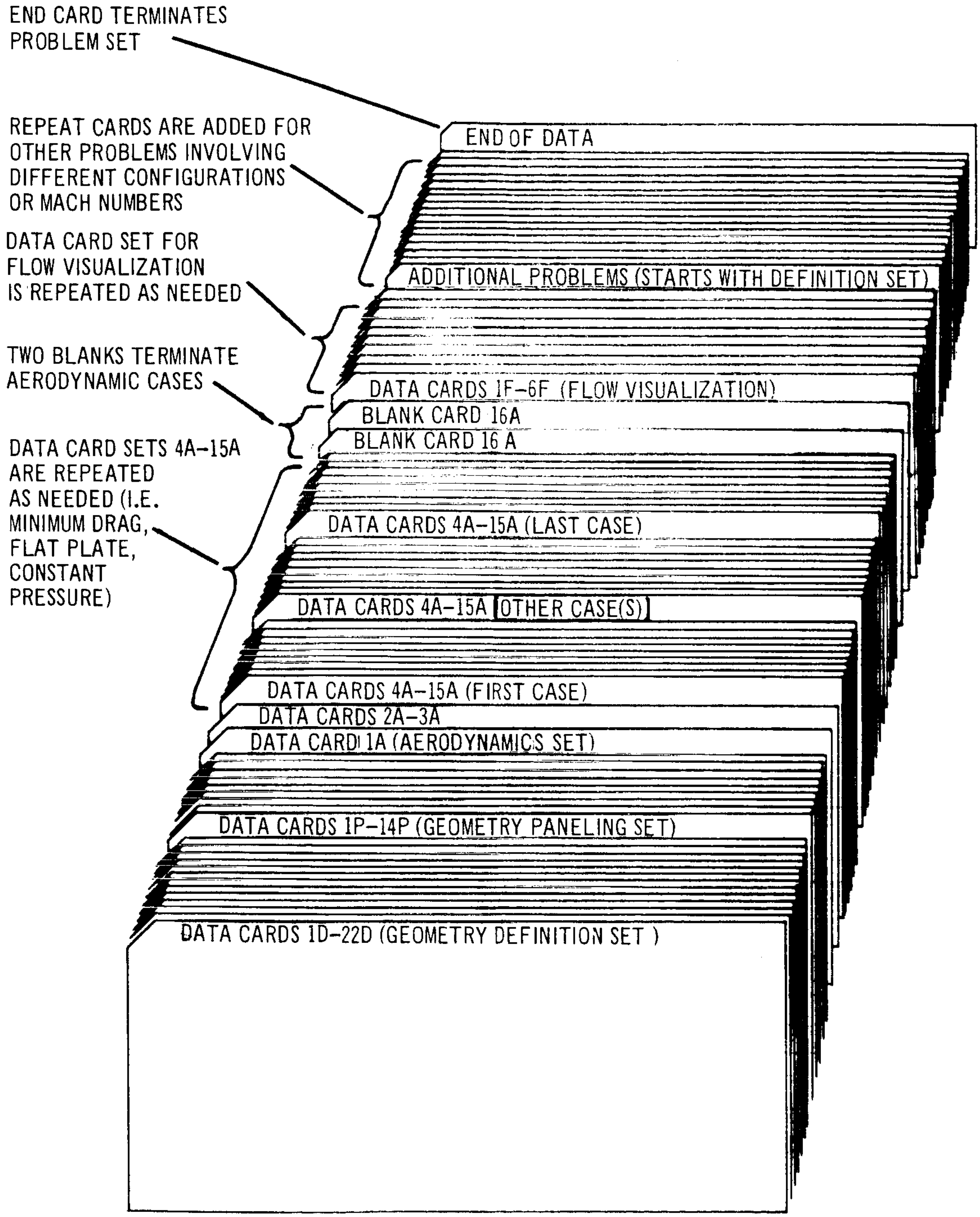
Note: For Aerodynamic section input data, only one of the following three card sets can be used per case:

1. Cards 10A-10AA (or -10AB). For CASE = 1. (design case), defines $\Delta C_p$ distribution.

2. Cards 11A-11AA, 12A-12AA (or -12AC). For CASE = 2. (analysis case), gives wing angle of attack, twist, and $\Delta z/\Delta x$ camber distribution.

3. Card 13A. For CASE = 3. (optimization case), gives constraints.

A typical stacking sequence of input data cards is illustrated in the following figure.



## GEOMETRY DEFINITION CARD SET

All geometry definition data, except title cards and literal statements, are punched in six-field, ten-digit format. A decimal point is required in each data field.

For a body-alone problem definition, Cards 10D through 20D are omitted. For a wing-alone problem definition, Cards 2D through 9D, 19D and 20D are omitted.

| | Column | Code | Explanation |
|---|---|---|---|
| Card 1D | 1-6 | DEFINE | Columns 1-6 contain the word DEFINE. |
| Card 2D | 1-4 | BODY | Columns 1-4 contain the work BODY. Card 2D is used only when a body or wing-body combination is defined. |
| Card 3D | 1-72 | TITLE | Any desired title. |
| Card 4D | 1-10 | BNS | Number of defining body stations. 2. $\leqslant$ BNS $\leqslant$ 50.. |
| | 11-20 | BTHETA | There is a defining-body cross section at each body station. This column gives number of points on each cross section. 3. $\leqslant$ BTHETA $\leqslant$ 10.. |
| | 21-30 | AXIS (1) | Y-coordinate of body definition axis. |
| | 31-40 | AXIS (2) | Z-coordinate of body definition axis. |
| | 41-50 | CHDB | Dimensional tolerance to be used in generating additional body-meridian line points between given stations. If CHDB $\leqslant$ 0. or if BNS < 4., no additional points will be generated. If 0. < CHDB $\leqslant$ 0.001, then a value of 0.001 will be used. |
| Card(s) 5D | 1-10 | $\theta_1$ | Array of angles ($\theta$), in degrees at each defining station. There must be exactly BTHETA angles $\leqslant$ 10, six per card. |
| | ⋮ | ⋮ | |
| (2 cards maximum) | 51-60 | $\theta_6$ | |
| | etc. | | |

| | Column | Code | Explanation |
|---|---|---|---|
| Card(s) 6D* (50 maximum) | 1-10 | STA | X-coordinate of body station. |
| | 11-20 | YZ(1) | $\Delta$ Y-increment added to body definition axis to establish a local origin from which all $\rho$, $\theta$ for this station are measured. |
| | 21-30 | YZ(2) | $\Delta$ Z-increment added to body definition axis to establish a local origin from which all $\rho$, $\theta$ for this station are measured. |
| | 31-40 | SCODE | = 0. this cross section is identical to previous section. |
| | | | = 1. this cross section is specified by BTHETA values of $\rho$ (on cards 7D). The $\theta$-array of card(s) 5D will be used. |
| | | | = 2. this cross section is a circle. (Radius given in columns 41-50.) |
| | | | = 3. this cross section is an ellipse. (Horizontal semi-axis is given in columns 41-50, the vertical in columns 51-60.) |
| | | | = 4. this cross section is circular (radius given in columns 41-50) with an angle array (on card(s) 8D) different from the $\theta$-array on card(s) 5D. This option allows local deviations in the meridian lines. |
| | | | = 5. this cross section is specified by a set of $\rho$ (on card(s) 7D) and by a nonstandard set of $\theta$ (on card(s) 8D). |
| | | | = 6. this cross section is given by a set of Y, Z pairs (on cards 9D). |

Note — If options 1, 4, 5, or 6 are designated, the added information card(s) 7D, 8D, or 9D must be inserted behind that station card 6D and before the next station card 6D.

*One card is needed for each defining station.

| | Column | Code | Explanation |
|---|---|---|---|
| | 41-50 | RAD(1) | Radius of section if SCODE = 2. or 4.. Horizontal semi-axis if SCODE = 3.. Not used otherwise. |
| | 51-60 | RAD(2) | Vertical semi-axis, if SCODE = 3.. Not used otherwise. |
| Card(s) 7D | 1-10<br>⋮<br>51-60 etc. | $\rho_1$<br>⋮<br>$\rho_6$ | A set of body radii $\rho$ if SCODE = 1. or 5.. There must be BTHETA $\leq$ 10 values of $\rho$. |
| Card(s) 8D (2 maximum per station) | 1-10<br>⋮<br>51-60 etc. | $\theta_1$<br>⋮<br>$\theta_6$ | A set of $\theta$ if SCODE = 4. or 5.. There must be BTHETA $\leq$ 10 values of $\theta$. |
| Card 9D (3 maximum per station) | 1-10 | $Y_1$ | Array of Y, Z coordinate pairs if SCODE = 6.. |
| | 11-20 | $Z_1$ | |
| | 21-30 | $Y_2$ | |
| | 31-40 | $Z_2$ | |
| | 41-50 | $Y_3$ | |
| | 51-60 etc. | $Z_3$ | |

Notes -

1. There is a maximum of three Card(s) 7D and/or Card(s) 8D per station.

2. For inputting a body meridian exactly coincident with a wing's intersection, it is recommended that the cross-section option SCODE = 6. be used and that CHDB = 0.. The Z-height of the intersection's meridian [Card(s) 9D] is held constant and at exactly the same height as the wing [Card(s) 17D]. The aft-most body-defining station must be a zero-radius circle.

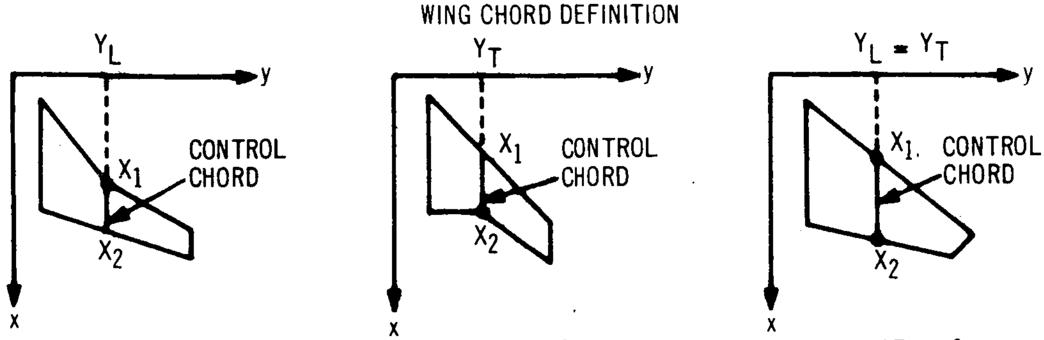
| | Column | Code | Explanation |
|---|---|---|---|
| Card 10D | 1-4 | WING | Columns 1-4 contain the word WING. This card is used whenever a wing is defined. For the case of a body alone, omit cards 10D through 19D. After reading a WING card, the program expects wing definition data. |
| Card 11D | 1-72 | TITLE | Any desired title. |
| Card 12D | 1-10 | PNLE | Number of corner or break points defining the planform leading edge. |
| | 11-20 | PNTE | Number of corner or break points defining the planform trailing edge. |
| | 21-30 | AFN | Number of planform control chords. AFN $\geq$ 2., including the wing-tip control chord. AFN must equal the larger of PNLE and PNTE; that is, each control chord must begin or end at a planform defining point. |
| | 31-40 | PLN | Number of constant percent chord lines used to form spanwise panel edges. Wing leading and trailing edges are counted in this number. |
| | 41-50 | WUL | = 1. |
| | 51-60 | CHD | Must be left blank. |
| Card 13D | 1-10 | PCODE | = 1. |
| | 11-20 | ACODE | = 1. |
| | 21-30 | EPS | Must be left blank. |
| Card(s) 14D | 1-10 | $X_1$ | Array of points defining the planform leading edge, arranged in order from inboard to outboard. There must be PNLE point pairs; three coordinates per card. |
| | 11-20 | $Y_1$ | |
| | 21-30 | $X_2$ | |
| | 31-40 | $Y_2$ | Notes - 1. For wing-body combinations, $X_1$ and $Y_1$ must lie inside the body so that an intersection can be calculated. |
| | 41-50 | $X_3$ | |

| | Column | Code | Explanation |
|---|---|---|---|
| | 51-60 | $Y_3$ etc. | 2. The X-axis defining these points originates at the nose of the configuration (for wing-body cases) as defined by the first Card 6D. |
| Card(s) 15D | 1-10 | $X_1$ | Array of points defining the planform trailing edge, arranged in order from inboard to outboard. There must be PNTE point pairs; three coordinates per card. |
| | 11-20 | $Y_1$ | |
| | 21-30 | $X_2$ | |
| | 31-40 | $Y_2$ | Note - For wing-body combinations, $X_1$ and $Y_1$ must lie inside the body so that an intersection can be calculated. |
| | 41-50 | $X_3$ | |
| | 51-60 | $Y_3$ etc. | |

Cards 16D — Cards 16D and 17D always occur in pairs (unless AFNU = 0. on card 16D) to define each wing control chord. There must be AFN $\geq$ 2. pairs of 16D and 17D cards.

| | Column | Code | Explanation |
|---|---|---|---|
| | 1-10 | AFK | Code to indicate how the control chord is oriented on the planform. See sketches below. |

WING CHORD DEFINITION




AFK = 1.

LEADING-EDGE POINT DEFINES CONTROL CHORD: The control chord leading point $X_1$ is at a planform leading-edge defining point.

AFK = 2.

TRAILING-EDGE POINT DEFINES CONTROL CHORD: The control chord trailing point $X_2$ is at a planform trailing-edge defining point.

AFK = 3.

LEADING- AND TRAILING-EDGE POINTS BOTH DEFINE CONTROL CHORD: The control chord leading edge $X_1$ is at a planform leading-edge defining point

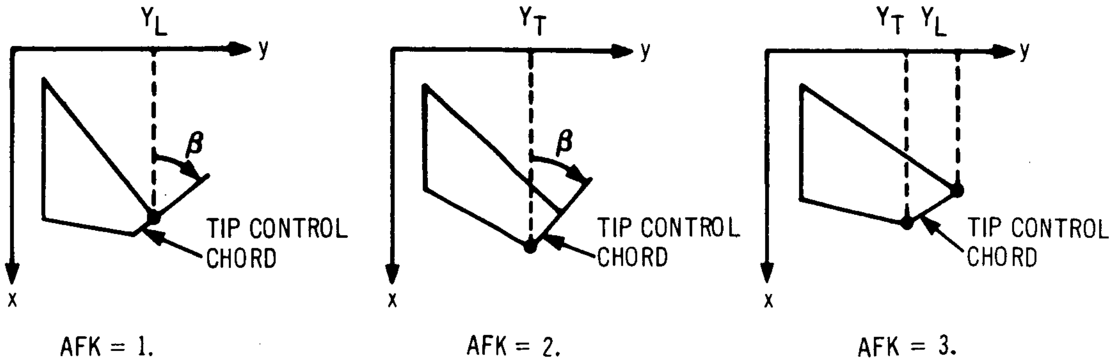
or

The control chord trailing point $X_2$ is at a planform trailing-edge defining point.

Both conditions must be satisfied at the tip.

| Column | Code | Explanation |
|---|---|---|

WINGTIP CONTROL CHORD DEFINITION (OBLIQUE)




| Column | Code | Explanation |
|---|---|---|
| | | Two of the three quantities $Y_L$, $Y_T$ or $\beta$ must be given. AFK indicates the appropriate pair. In summary: if AFK = 1., input $Y_L$ and $\beta$; if AFK = 2., input $Y_T$ and $\beta$; if AFK = 3., input $Y_T$ and $Y_L$. |
| 11-20 | BETA | The angle of yaw, $\beta$, in the diagram. If AFK = 1. or 2. and BETA = 0., the chord is streamwise. BETA is ignored if AFK = 3.. |
| 21-30 | $Y_L$ | Y-coordinate of leading edge of control chord. If AFK = 1. or 3., $Y_L$ is equal to the Y-coordinate of the corresponding planform leading-edge defining point. If AFK = 2., $Y_L$ is ignored. |
| 31-40 | $Y_T$ | Y-coordinate of trailing edge of control chord. If AFK = 2. or 3., $Y_T$ is equal to the Y-coordinate of the corresponding planform trailing-edge defining point. If AFK = 1., $Y_T$ is ignored. |
| 41-50 | AFNU | = 2. The height and control-chord true length are specified on the following card 17D. = 0. The previous 17D card values are used. Card 17D should not follow if AFNU = 0.. |

| | Column | Code | | Explanation |
|---|---|---|---|---|
| Card(s) 17D | 1-10 | $X_O$ | | = 0. |
| | 11-20 | $Z_O$ | | Z-coordinate at the leading edge of control chord. |
| | 21-30 | $X_C$ | | The control-chord true length. If $Z_O$ = $Z_C$ = 0, $X_C$ may be given an arbitrary length, which is then scaled by the program to make $X_C$ equal to the true chord length. |
| | 31-40 | $Z_C$ | | Z-coordinate of control chord at the trailing edge. |
| | Notes - | | 1. | $Z_O$ and $Z_C$ specify the height of the wing. $Z_O = Z_C$, always; furthermore, the height of each control chord, including the tip, must be the same. |
| | | | 2. | For wing-body configurations, $Z_O = Z_C$ must equal the height of the body meridian intended as the intersection meridian [Z-values, Card(s) 9D for SCODE = 6.] . |
| | | | 3. | The control-chord true length $X_C$ may be determined for AFK = 3. streamwise chords by subtracting the X-coordinates of the corresponding leading- and trailing-edge points (or by applying the right-triangle rule at an oblique chord).<br><br>For AFK = 1. or 2., interpolation must be done along the planform edge to locate a point corresponding to the chord's associated planform defining point. |
| Card(s) 18D | 1-10<br>⋮<br>51-60 etc. | $P_1$<br>⋮<br>$P_6$ | | Array of constant percent chord values corresponding to the panel spanwise edges. The leading-edge value $P_1$ = 0.. There are PLN values required with the last value (for the trailing edge) = 100. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card 19D | 1-3 | WBX | Columns 1-3 contain the letters WBX. This card indicates that a wing-body intersection is desired. For wing only or body alone cases, this card is omitted. |
| Card 20D | 1-10 | | = 1. linear interpolation used on body station perimeters to compute additional points between meridian lines in the wing intersection region.<br><br>= 2. biquadratic interpolation used on body station perimeters to compute additional points between meridian lines in the wing intersection region. |
| | 11-20 | | Dimensional intersection tolerance. Specifies the accuracy desired in locating wing-body intersection points. A value of 0.001 is suggested. |
| Card 21D | 1-5 | TDUMP | Columns 1-5 contain the letters TDUMP. This card is included if a dump of geometry definition and geometry transformation tapes is desired. See Appendix C for a detailed description of these tapes. |
| Card 22D | 1-6 | DEFEND | Columns 1-6 contain the word DEFEND. This card ends the definition set and must not be omitted. |

## GEOMETRY PANELING CARD SET

All paneling data, except title cards and literal statements, are punched in six-field, ten-digit format. A decimal point is required in each data field.

For body-alone case, cards 3P-14P are omitted.

For wing-alone case, cards 4P-7P are omitted.

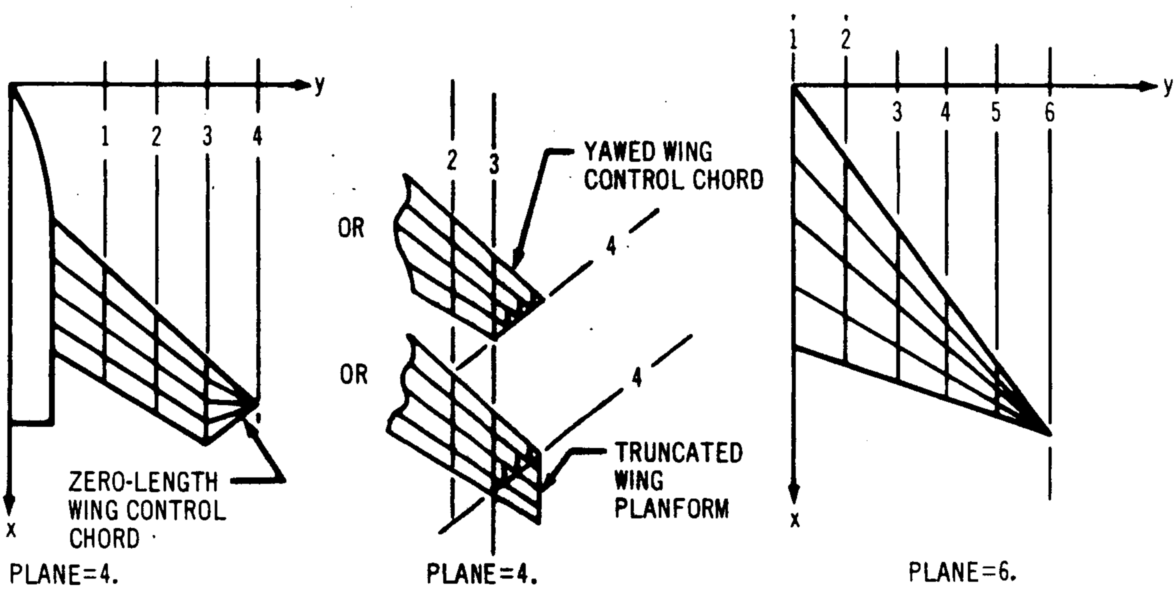
| | Column | Code | Explanation |
|---|---|---|---|
| Card 1P | 1-5 | PANEL | Columns 1-5 contain the word PANEL. This is the first card in the paneling link and must always follow the DEFEND card. |
| Card 2P | 1-10 | | The number of source control stations at which the radius for an equivalent body of circular cross section and the actual body station centroid height are computed. A maximum of 50 stations may be requested. It is recommended that 50 stations be used. The radius at each control station is used to determine the source strength necessary to simulate the body thickness. The camber height at each control station and the body angle of attack (Card 7A) are used to determine the doublet strength necessary to simulate body camber and angle of attack. |
| Note - | In a wing-alone problem, Card 2P is blank. | | |
| | 11-20 | | Dimensional tolerance applied to the additional points generated between meridian lines on the perimeter of body defining stations. This controls the area and centroid location calculations. A value of 0.001 is suggested. |
| | 21-30 | | This field contains an interpolation code. The program first determines an equivalent radius, R, at each body defining section, X, and then establishes an R vs. X array. Interpolation for additional radii at other stations is |

| | Column | Code | Explanation |
|---|---|---|---|
| | | | performed on this array. The same technique is used to determine centroid locations. |
| | | | = 1. linear interpolation for equivalent radii and centroid locations of the source control stations that are between body defining stations. |
| | | | = 2. biquadratic interpolation for equivalent radii and centroid locations at the source control stations that are between body defining stations. |
| | 31-40 | | = 1. linear interpolation between meridian line points on the body definition sections. |
| | | | = 2. if biquadratic interpolation is desired. |
| | 41-50 | | A dimensional tolerance value, E, such that if any equivalent radius length or centroid height, (z centroid), is less than E, its value will be set equal to zero. A value of 0.001 is suggested. |
| Card 3P | 1-10 | XPER | Fraction of local streamwise panel chord at which panel control point is located. 0. < XPER < 1.. |
| | | | Note - XPER = .95 for all cases discussed in this report. |
| | 11-20 | YPER | Fraction of local panel width at which panel control point is located. 0. < YPER < 1.. |
| | | | Note - YPER = 0. is a code used to locate the panel control point on the chord through the panel centroid. YPER = 0., for all cases discussed in this report. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card 4P | 1-10 | BODY PANEL | Columns 1-10 contain the words BODY PANEL. |
| Card 5P | 1-10 | PLNB* | Number of transverse body panel edges aft of wing trailing edge-body intersection ≤ 21. If PLNB = 0., omit card 6P. |
| | 11-20 | PLNW* | Number of transverse body panel edges within the wing body intersection region ≤ 16. |
| | 21-30 | TOLB | Slope tolerance on body secondary panel part leading edges. Panel parts with slopes $\left\lvert \frac{\Delta Y}{\Delta X} \right\rvert$ = TOLB in the local panel coordinate system) are eliminated. TOLB = 0.02 is suggested. |
| Card(s) 6P | 1-10<br>⋮<br>51-60<br>etc. | $XCEPTB_1$<br>⋮<br>$XCEPTB_6$ | x-values of transverse body panel edges aft of the wing trailing edge - body intersection. There are PLNB values. Omit this card(s) if PLNB = 0. |
| Card(s) 7P | 1-10<br>⋮<br>51-60<br>etc. | $CODEBW_1$<br>⋮<br>$CODEBW_6$ | Each field contains a number identifying those spanwise wing panel edges which continue around the body to form transverse body panel edges at the body intersection. The table must always start with the number 1. and terminate with the wing trailing-edge number. There are PLNW values. |
| Card 8P | 1-10 | WING PANEL | Columns 1-10 contain the words WING PANEL. |
| Card 9P | 1-10 | PLANE | Number of buttock lines which locate the streamwise wing panel edges specified by cards 10P and 11P.<br><br>Wing-alone problem: PLANE is the number of buttock lines locating the streamwise panel edges including both |

*(PLNB + PLNW) ≤ 21

| Column | Code | Explanation |
|---|---|---|
| | | the wing tip and centerline. |
| | | Wing-body problem: PLANE is the number of buttock lines locating the streamwise panel edges, but does not include the inboard edge located by the program at the wing-body intersection. PLANE ≥ 2.. See sketches below. |




| | | |
|---|---|---|
| 11-20 | OPTF | = 1. upper and lower airfoil ordinates are read in (cards 12P and 13P) at each wing buttock line passing through the panel centroids. If the wing is untwisted and has the same airfoil section from root to tip, only one airfoil table is necessary. The program will scale this table to fit the appropriate chord. |
| | | = 0. no tables are read in and the wing is a flat plate at zero incidence. |
| 21-30 | SNUM | Number of given airfoil ordinate tables. |
| | | = 0., only if OPTF = 0. |

| Column | Code | Explanation |
|---|---|---|
| | | = 1., same airfoil section from wing root to tip. |
| | | = (PLANE - 1), wing alone case airfoils specified. |
| | | = PLANE, wing-body case airfoils specified. |
| 31-40 | TOLW | Slope tolerance on wing secondary panel part leading edges. Panel parts with slopes $\left\lvert \frac{\Delta Y}{\Delta X} \right\rvert$ = TOLW are eliminated. TOLW = 0.01 is suggested. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card(s) 10P | 1-10<br>⋮<br>51-60<br>etc. | $YCEPT_1$<br>⋮<br>$YCEPT_6$ | Wing buttock line values at which streamwise panel edges are specified. There are (PLANE -1) values. The tip edge is specified on card 11P. |
| Card 11P | 1-10 | CPNT | Code indicating how the most outboard panel edge or wing tip is specified. |

Note - This card controls the outboard panel edge and in no way influences the spanwise edges which are established by the geometry definition. The outboard panel edge is usually made coincident with the definition wing tip, but it may be used to truncate the defined wing tip and the spanwise panel edges anywhere between the two outboard wing buttock lines specified by card 9P. If truncation is specified, the wing span and area are reduced.

= 0. X and Y coordinates of the wing tip leading and trailing edge are given. Use VALUE(1) through (4).

| Column | Code | Explanation |
|---|---|---|
| | | = 1. X and Y coordinates of the leading edge and the slope ($\Delta X/\Delta Y$) of the wing tip are given. Use VALUE(1), (2) and (5). |
| | | = 2. X and Y coordinates of the trailing edge and the slope ($\Delta X/\Delta Y$) of the wing tip are given. Use VALUE(3), (4) and (5). |
| 11-20 | VALUE(1) | X-coordinate of wing tip leading edge if CPNT = 0. or 1.. |
| 21-30 | VALUE(2) | Y-coordinate of wing tip leading edge if CPNT = 0. or 1.. |
| 31-40 | VALUE(3) | X-coordinate of wing tip trailing edge if CPNT = 0. or 2.. |
| 41-50 | VALUE(4) | Y-coordinate of wing tip trailing edge if CPNT = 0. or 2.. |
| 51-60 | VALUE(5) | wing tip slope, $\frac{\Delta X}{\Delta Y}$, if CPNT = 1. or 2.. |

Card(s) 12P

Card(s) 12P and Card(s) 13P give the SNUM set of airfoil coordinates. These card sets (12P and 13P) are always used in pairs to define each airfoil; the coordinates apply at the centroid buttock line of each chordwise row of wing panels. The inboard row is read in first, then the next row outboard, then the next, etc. out to and including the tip row.

Chordwise biquadratic interpolation is used between input points. Thus it is recommended that XNUM (1) be 25., or nearly 25.; input points should be concentrated around regions of rapidly varying thickness and/or camber, such as the airfoil leading edge.

The card sets (12P and 13P) are omitted if OPTF = 0.

| | Column | Code | Explanation |
|---|---|---|---|
| First Card | 1-10 | XNUM(1) | Number of points (X, Z coordinate pairs) in upper surface airfoil ordinate table. 4. $\leqslant$ XNUM(1) $\leqslant$ 25.. |

| | Column | Code | Explanation |
|---|---|---|---|
| Second Cards | 1-10 | $XFOIL_1$ | Upper surface airfoil ordinate table. |
| | 11-20 | $ZFOIL_1$ | Local X and Z coordinates are given from leading edge to trailing edge. That is, the first values of ZFOIL and XFOIL entered are always zero. If the wing has no twist, an unscaled set of ordinates may be given and the program will scale the airfoil to the local chord. The section has no twist, if and only if, |
| | . | . | |
| | . | . | |
| | . | . | |
| | 41-50 | $XFOIL_3$ | |
| | 51-60 | $ZFOIL_3$ | |
| | etc. | | $ZFOIL_{upper,\ trailing\ edge} = -(ZFOIL_{lower,T.E})$ |
| Cards 13P | | | |
| First Card | 1-10 | XNUM(2) | Number of points (X, Z coordinate pairs) in lower surface airfoil ordinate table. $4. \leqslant XNUM(2) \leqslant 25.$. |
| Second Cards | 1-10 | $XFOIL_1$ | Lower surface airfoil ordinate table. To input a completely uncambered section, it is recommended that XNUM(1)=XNUM(2) and that for I such that $1. \leqslant I \leqslant XNUM$, XFOIL(I), upper = XFOIL(I), lower and that ZFOIL(I), upper = - (ZFOIL(I), lower) |
| | 11-20 | $ZFOIL_1$ | |
| | . | . | |
| | . | . | |
| | . | . | |
| | 41-50 | $XFOIL_3$ | |
| | 51-60 | $ZFOIL_3$ | |
| | etc. | | |
| Card 14P | 1-6 | PANEND | Colums 1-6 contain the word PANEND. This card ends the paneling set and must be used whenever any paneling is performed. It is not needed for a body-alone problem. |

## AERODYNAMIC CARD SET

All aerodynamic data, except title cards and literal statements, are punched in seven-field, ten-digit format. A decimal point is required in each data field. Data Cards 1A and 2A are input only once for a given configuration and Mach number. The remaining aerodynamic data cards may be repeated as necessary to solve the selected aerodynamic cases.

| | Column | Code | Explanation |
|---|---|---|---|
| Card 1A | 1-11 | AERO-DYNAMIC | Columns 1-11 contain the word AERO-DYNAMIC. |
| Card 2A | Select and input one of the following data control cards. | | |
| | 1-7 | COMPUTE | The aerodynamic matrices and geometry data are not saved on tape; this card is used when a defined and paneled configuration is input. |
| | 1-9 | SAVE TAPE | The aerodynamic matrices and geometry data are saved on tape for repeated analysis. The body and wing thickness and camber from the geometry section are also saved for optional use. This card is used when a defined and paneled configuration is input. |
| | 1-8 | USE TAPE | A previously saved tape is to be used; no definition nor paneling data are input when this option is selected. |
| Card 3A | 1-10 | XMACH | Mach number. |
| | 11-20 | SYM | = 0. the aerodynamic problem solved is asymmetric about the vertical X-Z plane (image panels not included).<br><br>= 1. the aerodynamic problem solved is symmetric about the vertical X-Z plane (image panels included). |

| | Column | Code | Explanation |
|---|---|---|---|
| Card 4A | 1-72 | TITLE | Any desired title. |
| Card 5A | 1-10 | CASE | = 1. calculates wing twist and camber for a given $\Delta C_p$ distribution on wing where $\Delta C_p = C_{p\ lower} - C_{p\ upper}$<br><br>= 2. calculates pressure distribution over the configuration. Wing and body camber can be changed within this option.<br><br>= 3. optimizes wing twist and camber for minimum drag.<br><br>Note - For body-alone problems, only case = 2. option is available. |
| | 11-20 | CPCALC | = 0. $C_p$ calculations use linear equation, $C_p = -2u.$<br><br>= 1. $C_p$ calculations use nonlinear equation, $C_p = -2u + \beta^2 u^2 - v^2 - w^2$<br><br>= 2. $C_p$ calculations use the "exact" isentropic equation on the isolated body and the linear equation on the wing and body panels. The "exact" isentropic equation is |

$$C_p = \frac{2}{\gamma M^2} \left\{ \left[ 1 + \frac{\gamma - 1}{2} M^2 \left( 1 - (1 + u)^2 + v^2 + w^2 \right) \right]^{\gamma/\gamma - 1} - 1 \right\}$$

where M is the mainstream Mach number, U is the mainstream velocity and $\gamma$ (1.4) is the coefficient of specific heats.

| | Column | Code | Explanation |
|---|---|---|---|
| | 21-30 | POLAR | = 0. drag polar not requested.<br><br>= N. drag polar requested. A series of N incremental angles of attack is specified on Card(s) 15A. |
| | 31-40 | THICK | = 0. wing thickness pressures are not calculated.<br><br>= 1. wing thickness pressures are calculated. |

| | Column | Code | Explanation |
|---|---|---|---|
| | 41-50 | VOUT | = 0. the velocity components are not printed.<br>= 1. the velocity components are printed. |
| Card 6A | 1-10 | RFAREA | Half-wing reference area. If this field is left blank, the program sums the wing panel areas to obtain the reference area which is the half-wing exposed area. For the body-alone problem, a value must be input, or a unit area is used. |
| | 11-20 | XP | x-coordinate about which the pitching moments are computed. |
| | 21-30 | ZP | z-coordinate about which the pitching moments are computed. |
| | 31-40 | CBAR | = 0. Unit (1.) reference chord length to be used in pitching-moment calculations.<br>= CBAR, Reference chord length of CBAR to be used. |
| | 41-50 | SEMIS | = 0. Unit (1.) wing semispan to be used in spanwise $C_L$, $C_D$ calculations.<br>= SEMIS, Wing semispan of SEMIS to be used. |

Card 7A For configurations that include a body, the body angle of attack is specified on this card.

| Column | Code | Explanation |
|---|---|---|
| 1-10 | ARB | Body angle of attack in degrees. |

Card 8A For configurations that include a body, two options are available for specifying the body radii. The first word on the first card indicates the type of input. Omit this card set for wing-alone problems.

| Column | Code | Explanation |
|---|---|---|
| 1-5 | | Option A<br>= "GIVEN." The body radii to be used are those calculated in the geometry definition section. No additional cards are needed. |
| 1-72 | | Option B<br>= Identifying title. Any title calls the option of inputting body radii directly. The radii from the geometry section are superseded. |

| Column | Code | Explanation |
|---|---|---|

Card(s) 8AA (used only with Option B)

| | | |
|---|---|---|
| 1-10<br>11-20<br>.<br>.<br>.<br>61-70<br>etc. | $R_1$<br>$R_2$<br>.<br>.<br>.<br>$R_7$ | The body radii are input on these cards. They are input from nose to tail and apply at the source control stations generated by the paneling section. There are XNRX radii required (Card 2P, columns 1-10). |

Card 9A For configurations that include a body, two options are available for specifying body camber. The first word on the first card is the key to the type of input the program expects. Omit this card set for wing-alone problems.

| | | |
|---|---|---|
| 1-5 | GIVEN | Option A<br>Columns 1-5 contain the word GIVEN. |
| 7-80 | Any additional identifying symbols | The program takes the body camber as that calculated in the geometry definition section. No additional cards are necessary for this option. |
| 1-80 | Any identifying symbols | Option B<br>The first card contains any arbitrary identifying symbols (other than GIVEN or CONSTANT as the first word) to describe the body camber. The program expects more cards immediately to specify the body camber. |

Card(s) 9AA (used only with Option B)

| | | |
|---|---|---|
| 1-10<br>11-21<br>.<br>.<br>.<br>61-70<br>etc. | $Z_1$<br>$Z_2$<br>.<br>.<br>.<br>$Z_7$ | The body camber is input on these cards as Z-heights of body cross sections at the source control stations. The values are input from nose to tail; there are XNRX values required (Card 2P, columns 1-10). |

Card 10A Calculates wing twist and camber for a given wing $\Delta C_p$ distribution (CASE = 1., field 1 of Card 5A). Two options are available for specifying the $\Delta C_p$ distribution. These options are selected by the first word on the first card of this set. Omit this set for body-alone problems or CASE = 2. or 3.

| | | |
|---|---|---|
| 1-8 | CONSTANT | Option A<br>Columns 1-8 contain the word CONSTANT. |
| 9-80 | Any additional identifying symbols | This option restricts the wing to have a constant $\Delta C_p$ distribution. This constant value is specified on the following card. Recall that $\Delta C_p = C_{p\ lower} - C_{p\ upper}$ |

| Column | Code | Explanation |
|---|---|---|
| 1-80 | Any identifying symbols | Option B<br>The first card contains appropriate identifying symbols (other than GIVEN or CONSTANT as the first word) to select Option B. $\Delta C_p$ for each panel is specified on the following card set. |

Card 10AA (used only with Option A)

| | | |
|---|---|---|
| 1-10 | $\Delta C_p$ | $\Delta C_p$ for Option A. |

Card(s) 10AB (used only with Option B)

| | | |
|---|---|---|
| 1-10<br>.<br>.<br>.<br>.<br>61-70<br>etc. | $\Delta C_{p_1}$<br>.<br>.<br>.<br>.<br>$\Delta C_{p_7}$ | $\Delta C_p$'s for Option B. This array must be ordered starting with the inboard panel at the leading edge and running aft to the trailing edge, then proceeding outboard to the tip in the same manner. There must be the same number of values as there are wing panels. Values apply at panel centroids. |

Card 11A For configurations that include a wing and for which CASE (Card 5A) = 2, the wing angle of attack and an optional twist distribution are indicated on these cards.

| | | |
|---|---|---|
| 1-10 | ARW | Wing angle of attack in degrees (relative to body axis if wing-body configuration) |
| 11-20 | TWIST | = 0. No twist distribution to be given.<br>= 1. Twist distribution to be specified on following cards. |

Card(s) 11AA (used only if TWIST = 1.)

| | | |
|---|---|---|
| 1-10<br>11-20<br>.<br>.<br>61-70<br>etc. | $ARWT_1$<br>$ARWT_2$<br>.<br>.<br>$ARWT_7$ | Twist angle for successive wing panel columns; $ARWT_1$ applies to the inboard-most column. |

Note - Card 11AA is repeated until angle for each column of wing panels is given. Do not use Card 11AA if TWIST (Card 11A) = 0.

Card 12A Calculates the pressure distribution over the configuration (CASE = 2., field 1 of Card 5A). Three options are available for specifying the camber shape of the wing. The options are selected by the first word on the first card of this set. Omit this set for body-alone problems or CASE = 1. or 3.

| Column | Code | Explanation |
|---|---|---|
| | | Option A |
| 1-8 | CONSTANT | Columns 1-8 contain the word CONSTANT. This option restricts the wing camber |
| 9-80 | Any other identifying symbols | shape to have a constant slope for each wing panel. This constant value is specified on the following card. |
| | | Option B |
| 1-5 | GIVEN | The wing camber shape is specified by the input geometry. The panel slopes |
| 7-80 | Any other identifying symbols | used are those generated in the paneling section of the program. In this case, no more cards are needed. |
| | | Option C |
| 1-80 | Any identifying symbols | Appropriate identifying symbols (other than GIVEN or CONSTANT as the first word) on the first card of this set are used to select this option. The wing camber shape is specified by a slope for each panel. Other cards must be input which contain the slope values. |

Card 12AA (used only with Option A)

| | | |
|---|---|---|
| 1-10 | ALPHAC | The constant wing panel $\Delta z/\Delta x$ slope for Option A. |

There is no auxiliary card for Option B.

Card(s) 12AC (used only with Option C)

| | | |
|---|---|---|
| 1-10 | $ALPHAC_1$ | Wing panel $\Delta z/\Delta x$ slopes for Option C. The array must be ordered starting with |
| . | . | the inboard panel at the leading edge and |
| . | . | running aft to the trailing edge, then |
| 61-70 | $ALPHAC_7$ | proceeding outboard to the tip in the same |
| etc. | | manner. There must be the same number of values as there are wing panels. Values apply to panel control points. |

Card 13A Optimizes wing twist and camber for minimum drag (CASE = 3., field 1 of Card 5A). Two options are available. The first option optimizes the wing for a given wing-lift constraint, and the second option optimizes the wing for both the wing-lift and center-of-pressure constraints. Only one data card is required. Omit this card for a body-alone problem or CASE = 1. or 2.

| | | |
|---|---|---|
| 1-10 | CONSNT | = 0. the wing is optimized for minimum drag with a wing-lift constraint. |

| | Column | Code | Explanation |
|---|---|---|---|
| | | | = 1. the wing is optimized for minimum drag with both wing-lift constraint and x-coordinate of the center-of-pressure constraint. |
| | 11-20 | CLBAR | Wing-lift-coefficient constraint. |
| | 21-30 | XCPBAR | The x-coordinate of the wing center-of-pressure constraint. If the center of pressure is not constrained, omit this field. |

Card 14A For configurations that include a wing and for which THICK (Card 5A) = 1., two options are available for specifying the wing thickness distribution. The first word on the first card indicates the type of input. Omit this card and card(s) 14AB if thickness effects are not requested (THICK = 0).

| | | |
|---|---|---|
| | | Option A |
| 1-5 | | = "GIVEN." The wing thickness distribution to be used is that computed in the geometry paneling section. No more cards are needed. |
| | | Option B |
| 1-72 | | = Identifying title. The wing thickness distribution is input on the following cards. |

Card(s) 14AB (use only with Option B)

| | | |
|---|---|---|
| 1-10<br>11-20<br>21-30<br>.<br>.<br>.<br>61-70 | $ALPHAT_1$<br>$ALPHAT_2$<br>$ALPHAT_3$<br>.<br>.<br>.<br>$ALPHAT_7$ | Wing thickness $\Delta z/\Delta x$ slopes. For the purpose of these cards, wing thickness is the distance from the airfoil camber line to either airfoil surface. A leading-edge thickness slope (less than the tangent of the Mach angle) is given for the inboard wing-panel column and followed by thickness slopes for each panel in that column of the wing panels. This is repeated for each wing-panel column. The total number of input wing-thickness slopes equals the number of wing panels plus the number of panel columns. |

| | | | |
|---|---|---|---|
| Card(s) 15A | 1-10 | DADEG | If drag polar is requested (POLAR = N. on Card 5A), incremental angles of attack (in degrees) are specified on Card(s) 15A. A terminating blank card is used. Omit this card set if Polar option is not selected (POLAR = 0.). |

## FLOW VISUALIZATION CARD SET

All flow visualization data, except title cards and literal statements, are punched in seven-field, ten-digit format. A decimal point is required in each data field. This card set instruction applies to any type of configuration and any type of aerodynamic analysis. Only the last angle of attack of the last aerodynamic analysis is visualized.

| | Column | Code | Explanation |
|---|---|---|---|
| Card 1F | 1-8 | FLOW VIZ | Columns 1-8 contain the word FLOW VIZ. This card calls the Flow Visualization section of the program. |
| Card 2F | 1-10 | CPCALC | = 0. The linear $C_p$ equation is used to calculate field pressure points. |
| | | | = 1. The nonlinear $C_p$ equation is used. |
| | | | = 2. The "exact" isentropic equation is used. |
| | | | The $C_p$ equations are defined for Card 5A of the aerodynamic set. |
| | 11-20 | CAMN | = 1. Visualization calculation of only the thickness effects of the wing and/or body. No camber, twist, angle-of-attack, or interference effects are computed. If lifting effects are not of interest, selecting this option will save computer time. |
| | | | = 0. Visualization calculation takes account of all effects included in the analysis. |
| | 11-21 | POP | = 0. No special print option. |
| | | | $\neq$ 0. Prints velocity components due to various singularities. |

NOTE: The three modes of flow visualization (grids, streamlines, and points) may be called in any order and any number of times by inputting the appropriate literal card.

| | Column | Code | Explanation |
|---|---|---|---|
| Card 3F | 1-5 | GRIDS | Columns 1-5 contain the word GRIDS. This card calls the grid mode of flow visualization. |
| Card 3FA | 1-10 | XNG | The number of two-dimensional or three-dimensional grid structures; any number may be requested. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card(s) 16A | 1-72 | (blank) | (This is a blank card.) To conduct an additional aerodynamic analysis, begin the new set of aerodynamic data with Card 4A. To terminate the data of the last analysis or to call the flow visualization section, place two Card(s) 16A after the terminating polar card. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card 3FB | 1-10<br>11-20<br>21-30 | XO<br>YO<br>ZO | The rectangular coordinates of the origin of the grid structure. |
| | 31-40<br>41-50<br>51-60 | XA<br>YA<br>ZA | The rectangular coordinates of a point which, together with the origin (above), defines an axis of a skew grid; the positive direction is from the origin to point A.<br><br>Enter values only if a skew grid is called; AMD (Card 3FC) = 4. |
| | 61 | LIT | = X<br>= Y<br>= Z Select and input the letter of the coordinate axis which is used together with the skew axis defined above to define the two-dimensional skew grid.<br><br>Enter a letter only if AMD = 4. |
| Card 3FC | 1-10<br>11-20<br>21-30 | DX<br>DY<br>DZ | Values of increments which position points on the grid. If a two-dimensional grid is desired, set one of the increments to zero and the corresponding increment count (next three fields) to 1. |
| | 31-40<br>41-50<br>51-60 | XN<br>YN<br>ZN | The respective number of increments which position points on the grid.<br>If AMD = 1., YN $\times$ ZN $\leq$ 500.<br>If AMD = 2., XN $\times$ ZN $\leq$ 500.<br>If AMD = 3., XN $\times$ ZN $\leq$ 500. |
| | 61-70 | AMD | This code controls the format in which the data is output.<br><br>= 1. Data is output for all the (Y, Z) points at a given X-value; then the data for the next X-value, etc. Always select this code value when DX = 0.<br><br>= 2. Data is output for all the (X, Z) points at a given Y-value; then the data for the next Y-value, etc. Always select this code value when DY = 0. |

| Column | Code | Explanation |
|---|---|---|
| | | = 3. Data is output for all the (X, Y) points at a given Z-value; then the data for the next Z-value, etc. Always select this code value when DZ = 0. |
| | | = 4. A skew grid has been specified for the plane defined by LIT-axis and the line from the origin through point A. The following notes apply. |

Notes— 1. Increments and the number of increments determining points on a skew grid are input in the following way: The axis specified by LIT is input as if for a rectangular array. One of the other two axes is input with zero (0.) increment size and number. The remaining axis is input as if for a rectangular array also, but its graduations apply to the skew axis; planes are established normal to the remaining axis through each of its graduations. Where these planes intersect the skew axis, a skew-axis graduation is established.

2. Repeat Card 3FB and Card 3FC for each grid called. That is, there must be KNG sets of Cards 3FB-3FC.

| | Column | Code | Explanation |
|---|---|---|---|
| Card 4F | 1-11 | STREAMLINES | This word appears in columns 1-11 and calls the streamlines option. |
| Card 4FA | 1-10 | DXMAX | The maximum permissible step size for the streamline's integrator. A value of 10. × XDELT is recommended. |
| | 11-20 | DXMIN | The minimum permissible step size for the streamline's integrator. DXMIN > 0., always. A value of 0.1 × XDELT is recommended. |
| | 21-30 | PRINT | The result is a series of points in rectangular coordinates (X, Y, Z) from XMIN through point S to XMAX. The values of X are determined by this code. (See Card 4FC.) |
| | | | = 0. Points are printed out in even increments of XDELT. |
| | | | = 1. Points are generated by the integrator and are printed out directly. |

| | Column | Code | Explanation |
|---|---|---|---|
| | Note— | If printout with PRINT = 1. shows that DXMIN is reached as an actual interval in printing out the results, the streamline may not be accurate. To ensure accuracy, reduce DXMIN and XDELT and repeat the calculation. | |
| Card 4FB | 1-10 | XNS | The number of streamlines to be specified; any number may be requested. |
| Card(s) 4FC | 1-10<br>11-20<br>21-30 | XS<br>YS<br>ZS | The streamlines' starting point. The streamline begins on the XMIN plane, proceeds downstream, passes through point S, and continues to the XMAX plane. |
| | 31-40<br>41-50 | XMIN<br>XMAX | The farthest point upstream and downstream to which the streamline will be calculated. The relationship XMIN $\leqslant$ XS $\leqslant$ XMAX must be observed. |
| | 51-60 | XDELT | The initial step size for the streamline's integration. For configurations that include a wing, a value of 0.1 $\times$ (mean chord length) is recommended. For body-alone analyses, a value of 0.1 $\times$ (body length) is recommended. For streamlines passing very near the configuration, a smaller value should be used. DXMAX > XDELT > DXMIN, always.<br><br>There must be exactly XNS number of Card(s) 4FC. |
| Card 5F | 1-6 | POINTS | Columns 1-6 contain the word POINTS. This card calls the points mode of the flow visualization. |
| Card 5FA | 1-10 | XP | The number of points requested; any number may be requested. |
| Card(s) 5FB | 1-10<br>11-20<br>21-30 | XF<br>YF<br>ZF | Rectangular coordinates of point; there must be XP number of Card(s) 5FB. |

| | Column | Code | Explanation |
|---|---|---|---|
| Card 6F | 1-8 | VIZEND | Columns 1-6 contain the word VIZEND. This card terminates the Flow Visualization section. The flow visualization calculations may be continued after the above data has been calculated and this card has been entered, by inputting Card 1F again and continuing the data. |
| Terminal Card | 1-11 | END OF DATA | If no further analysis is required, the run is terminated by an END OF DATA card. |

### 3.4 Sample Input and Output Data

**SEVEN FIELD, TEN DIGIT CRD FORMAT**

| 1–10 | 11–20 | 21–30 | 31–40 | 41–50 | 51–60 | 61–72 | IDENT 73–80 |
|---|---|---|---|---|---|---|---|
| DEFINE | | | | | | | 1D |
| BODY | | | | | | | 2D |
| TR-805/EQUIVALENT/BODY | | | | | | | 3D |
| 24. | 8. | | | | | | 4D |
| 0. | 25. | 50. | 75. | 102.19 | 130. | | 5D |
| 155. | 180. | | | | | | 5D |
| 0. | 0. | 0. | 2. | 0. | | | 6D |
| 1.5 | 0. | 0. | 2. | .270 | | | 6D |
| 3. | 0. | 0. | 2. | .4464 | | | 6D |
| 4.5 | 0. | 0. | 2. | .5943 | | | 6D |
| 6. | 0. | 0. | 2. | .7234 | | | 6D |
| 7.5 | 0. | 0. | 2. | .837 | | | 6D |
| 9. | 0. | 0. | 2. | .9364 | | | 6D |
| 10.5 | 0. | 0. | 2. | 1.0223 | | | 6D |
| 12. | 0. | 0. | 2. | 1.0936 | | | 6D |
| 13.5 | 0. | 0. | 2. | 1.1479 | | | 6D |
| 15. | 0. | 0. | 2. | 1.184 | | | 6D |
| 15.1 | 0. | 0. | 0. | | | | 6D |
| 17.5 | 0. | 0. | 0. | | | | 6D |
| 20. | 0. | 0. | 0. | | | | 6D |
| 22.5 | 0. | 0. | 0. | | | | 6D |
| 25. | 0. | 0. | 0. | | | | 6D |
| 27.5 | 0. | 0. | 0. | | | | 6D |
| 29.9 | 0. | 0. | 0. | | | | 6D |
| 31. | 0. | 0. | 2. | 1.174 | | | 6D |
| 32. | 0. | 0. | 2. | 1.154 | | | 6D |
| 33. | 0. | 0. | 2. | 1.124 | | | 6D |
| 34. | 0. | 0. | 2. | 1.084 | | | 6D |
| 35. | 0. | 0. | 2. | 1.034 | | | 6D |
| 36. | 0. | 0. | 2. | 1. | | | 6D |
| WING | | | | | | | 10D |
| TR-805 WING (TWP = -.25) | | | | | | | 11D |
| 2. | 3. | 4. | 11. | 1. | 0. | | 12D |
| 1. | 1. | 0. | | | | | 13D |
| 10.67 | 0. | 40.27 | 10.774 | | | | 14D |
| 19.88 | 0. | 43.518 | 8.603 | 40.27 | 10.774 | | 15D |
| 3. | 0. | 0. | 0. | 2. | | | 16D |
| 0. | -.25 | 9.21 | -.25 | | | | 17D |
| 1. | 0. | 5. | 0. | 0. | | | 16D |
| 3. | 0. | 8.603 | 8.603 | 2. | | | 16D |
| 0. | -.25 | 9.21 | -.25 | | | | 17D |
| 3. | 0. | 10.774 | 10.774 | 2. | | | 16D |
| 0. | -.25 | 0. | -.25 | | | | 17D |
| 0. | 10. | 20. | 30. | 40. | 50. | | 18D |
| 60. | 70. | 80. | 90. | 100. | | | 18D |
| WBX | | | | | | | 19D |
| 2. | .0001 | | | | | | 20D |
| DEFEND | | | | | | | 22D |
| PANEL | | | | | | | 1P |
| 50. | 0. | 1. | 1. | .001 | | | 2P |
| .95 | 0. | | | | | | 3P |
| BODY PANEL | | | | | | | 4P |
| 4. | 11. | .02 | .01 | | | | 5P |
| 25. | 27. | 29.5 | 32.415 | | | | 6P |

TITLE BOEING WING-BODY MODEL NAME DATE PAGE OF

## SEVEN FIELD, TEN DIGIT CRD FORMAT

| 1–10 | 11–20 | 21–30 | 31–40 | 41–50 | 51–60 | 61–70 | IDENT 73–80 |
|---|---|---|---|---|---|---|---|
| 1. | 2. | 3. | 4. | 5. | 6. | | 7P |
| 7. | 8. | 9. | 10. | 11. | | | 7P |
| WING PANEL | | | | | | | 8P |
| 10. | 1. | 1. | .05 | | | | 9P |
| 1.9846 | 2.8119 | 3.6392 | 4.4665 | 5.2938 | 6.1211 | | 10P |
| 6.9484 | 7.7757 | 8.603 | | | | | 10P |
| 0. | 40.27 | 10.774 | 40.27 | 10.774 | | | 11P |
| 15. | | | | | | | 12P |
| 0. | 0. | .025 | .0108 | .05 | .01556 | | 12P |
| .1 | .02155 | .15 | .02548 | .2 | .02832 | | 12P |
| .3 | .03215 | .4 | .0342 | .5 | .03385 | | 12P |
| .6 | .03112 | .7 | .02599 | .8 | .01915 | | 12P |
| .9 | .01084 | .95 | .00616 | 1. | 0. | | 12P |
| 15. | | | | | | | 13P |
| 0. | 0. | .025 | -.00257 | .05 | -.00325 | | 13P |
| .1 | -.00428 | .15 | -.0051 | .2 | -.00581 | | 13P |
| .3 | -.00701 | .4 | -.00735 | .5 | -.0065 | | 13P |
| .6 | -.00469 | .7 | -.00239 | .8 | -.00068 | | 13P |
| .9 | 0. | .95 | 0. | 1. | 0. | | 13P |
| PANEND | | | | | | | 14P |
| AERODYNAMIC | | | | | | | 1A |
| SAVE TAPE | | | | | | | 2A |
| 1.8 | | | | | | | 3A |
| TR 805/EQUIVALENT/BODY | | | | | | | 4A |
| 2. | 0. | 0. | 1. | 1. | | | 5A |
| 89.375 | 0. | 0. | | | | | 6A |
| 6. | | | | | | | 7A |
| GIVEN BODY RADII | | | | | | | 8A |
| GIVEN BODY CAMBER | | | | | | | 9A |
| 0. | 0. | | | | | | 11A |
| CONSTANT WING CAMBER | | | | | | | 12A |
| 0. | | | | | | | 12AA |
| GIVEN WING THICKNESS | | | | | | | 14A |
| | | | | | | | 16A |
| | | | | | | | 16A |
| FLOW VIZ | | | | | | | 1F |
| 0. | 0. | | | | | | 2F |
| POINTS | | | | | | | 5F |
| 9. | | | | | | | 5FA |
| 24. | 4. | -1. | | | | | 5FB |
| 24. | 4. | -.5 | | | | | 5FB |
| 24. | 4.3535 | -.6465 | | | | | 5FB |
| 24. | 4.5 | -1. | | | | | 5FB |
| 24. | 4.3535 | -1.3535 | | | | | 5FB |
| 24. | 4. | -1.5 | | | | | 5FB |
| 24. | 3.6465 | -1.3535 | | | | | 5FB |
| 24. | 3.5 | -1. | | | | | 5FB |
| 24. | 3.6465 | -.6465 | | | | | 5FB |
| GRIDS | | | | | | | 3F |
| 2. | | | | | | | 3FA |
| 30. | 1.2 | -1.2 | | | | | 3FB |
| .5 | 1. | .5 | 10. | 3. | 2. | 2. | 3FC |
| 14.90 | 0. | -1000. | | | | | 3FB |
| 1. | 200. | 0. | 70. | 1. | 0. | .2 | 3FC |

TITLE BOEING WING-BODY MODEL NAME DATE PAGE OF

AD 3779 R1 6-8300

## SEVEN FIELD, TEN DIGIT CRD FORMAT

| 1 10 | 11 20 | 21 30 | 31 40 | 41 50 | 51 60 | 61 69 | 70 | 71 | 72 | IDENT 80 |
|---|---|---|---|---|---|---|---|---|---|---|
| STREAMLINES | | | | | | | | | | 4F |
| 5. | .01 | 1. | | | | | | | | 4FA |
| 2. | | | | | | | | | | 4FB |
| 28. | 5. | .25 | 40. | 20. | .1 | | | | | 4FC |
| 28. | 5. | -.5 | 40. | 20. | .2 | | | | | 4FC |
| VIZEND | | | | | | | | | | 6F |
| END OF DATA | | | | | | | | | | |

TITLE BOEING WING-BODY MODEL NAME DATE PAGE OF

AD 3778 R1 6-8200

Sample Data Printout

The computer output for a sample case is given on the following pages:

```
PROGRAM FOR ANALYSIS AND DESIGN OF SUPERSONIC WING-BODY COMBINATIONS   MAY 20, 1967

DATA CARDS ARE LISTED BELOW

DEFINE                                                                  1D
BODY                                                                    2D
TR-805/EQUIVALENT/BODY                                                  3D
24.        8.                                                           4D
0.         25.        50.        75.        102.19     130.             5D
155.       180.                                                         5D
0.         0.         0.         2.         0.                          6D
1.5        0.         0.         2.         .270                        6D
3.         0.         0.         2.         .4464                       6D
4.5        0.         0.         2.         .5943                       6D
6.         0.         0.         2.         .7234                       6D
7.5        0.         0.         2.         .837                        6D
9.         0.         0.         2.         .9364                       6D
10.5       0.         0.         2.         1.0223                      6D
12.        0.         0.         2.         1.0936                      6D
13.5       0.         0.         2.         1.1479                      6D
15.        0.         0.         2.         1.184                       5D
15.1       0.         0.         0.                                     6D
17.5       0.         0.         0.                                     6D
20.        0.         0.         0.                                     6D
22.5       0.         0.         0.                                     6D
25.        0.         0.         0.                                     6D
27.5       0.         0.         0.                                     6D
29.9       0.         0.         0.                                     6D
31.        0.         0.         2.         1.174                       6D
32.        0.         0.         2.         1.154                       6D
33.        0.         0.         2.         1.124                       6D
34.        0.         0.         2.         1.084                       6D
35.        0.         0.         2.         1.034                       6D
36.        0.         0.         2.         1.                          6D
WING                                                                    10D
TR-805 WING(TWP=-.25)                                                   11D
2.         3.         4.         11.        1.
1.         1.         0.                                                13D
10.67      0.         40.27      10.774                                 14D
19.88      0.         43.518     8.603      40.27      10.774
3.         0.         0.         0.         2.                          16D
0.         -.25       9.21       -.25                                   17D
1.         0.         5.         0.         0.                          16D
3.                    8.603      8.603      2.
0.         -.25       9.21       -.25
3.                    10.774     10.774     2.
0.         -.25       0.         -.25
0.         10.        20.        30.        40.        50.              18D
60.        70.        80.        90.        100.                        18D
WBX                                                                     19D
2.         .0001                                                        20D
DEFEND                                                                  22D
PANEL                                                                   1P
50.        0.         1.         1.         .001                        2P
.95        0.                                                           3P
BODY PANEL                                                              4P
4.         11.        .02        .01                                    5P
25.        27.        29.5       32.415                                 6P
1.         2.         3.         4.         5.         6.               7P
```

```
7.         8.         9.         10.        11.                                        7P
WING PANEL                                                                             8P
10.        1.         1.         .05                                                   9P
1.9846     2.8119     3.6392     4.4665     5.2938     6.1211
6.9484     7.7757     8.603
0.         40.27      10.774     40.270     10.774                                     11P
15.                                                                                    12P
0.         0.         .025       .0108      .05        .01556                          12P
.1         .02155     .15        .02548     .2         .02832                          12P
.3         .03215     .4         .0342      .5         .03385                          12P
.6         .03112     .7         .02599     .8         .01915                          12P
.9         .01084     .95        .00616     1.         0.                              12P
15.                                                                                    13P
0.         0.         .025       -.00257    .05        -.00325                         13P
.1         -.00428    .15        -.0051     .2         -.00581                         13P
.3         -.00701    .4         -.00735    .5         -.0065                          13P
.6         -.00469    .7         -.00239    .8         -.00068                         13P
.9         0.         .95        0.         1.         0.                              13P
PANEND                                                                                 14P
AERODYNAMIC                                                                            1A
SAVE TAPE
1.8        1.
TR-805/EQUIVALENT/BODY
2.         0.         0.         1.         1.
89.375     0.         0.
6.
GIVEN BODY RADII
GIVEN BODY CAMBER
0.         0.
CONSTANT WING CAMBER
0.
GIVEN WING THICKNESS


FLOW VIZ
0.         0.
POINTS
9.
24.        4.         -1.
24.        4.         -.5
24.        4.3535     -.6465
24.        4.5        -1.
24.        4.3535     -1.3535
24.        4.         -1.5
24.        3.6465     -1.3535
24.        3.5        -1.
24.        3.6465     -.6465
GRIDS
2.
30.        1.2        -1.2
.5         1.         .5         10.        3.         2.         2.
1490.      0.         -1000.
1.         200.       0.         70.        1.         0.         2.
STREAMLINES
5.         .01        1.
3.
28.        5.         .25        40.        20.        .1
28.        5.         -.5        40.        20.        .2
15.4       1.5        0.         30.        0.         .1
VIZEND
END OF DATA

     A TOTAL OF   116  DATA CARDS WERE READ
```

DEFINE 1D TIME 01.572 052067

BODY 2D

BODY DEFINITION MAY 20, 1967

TR-805/EQUIVALENT/BODY

THERE ARE 24 STATIONS AND 8 MERIDIAN LINES

THE MAIN BODY AXIS (PARALLEL TO THE X AXIS) IS LOCATED AT
Y = -0.0000 Z = -0.0000

CHORD-HEIGHT TOLERANCE = -0.0000

BODY STATION X = 0.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0000 | 25.0000 | 0.0000 | 0.0000 |
| 3 | 0.0000 | 50.0000 | 0.0000 | 0.0000 |
| 4 | 0.0000 | 75.0000 | 0.0000 | 0.0000 |
| 5 | 0.0000 | 102.1900 | 0.0000 | -0.0000 |
| 6 | 0.0000 | 130.0000 | 0.0000 | -0.0000 |
| 7 | 0.0000 | 155.0000 | 0.0000 | -0.0000 |
| 8 | 0.0000 | 180.0000 | 0.0000 | -0.0000 |

BODY STATION X = 1.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.2700 | 0.0000 | 0.0000 | 0.2700 |
| 2 | 0.2700 | 25.0000 | 0.1141 | 0.2447 |
| 3 | 0.2700 | 50.0000 | 0.2068 | 0.1736 |
| 4 | 0.2700 | 75.0000 | 0.2608 | 0.0699 |
| 5 | 0.2700 | 102.1900 | 0.2639 | -0.0570 |
| 6 | 0.2700 | 130.0000 | 0.2068 | -0.1736 |
| 7 | 0.2700 | 155.0000 | 0.1141 | -0.2447 |
| 8 | 0.2700 | 180.0000 | 0.0000 | -0.2700 |

BODY STATION X = 3.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.4464 | 0.0000 | 0.0000 | 0.4464 |
| 2 | 0.4464 | 25.0000 | 0.1887 | 0.4046 |
| 3 | 0.4464 | 50.0000 | 0.3420 | 0.2869 |
| 4 | 0.4464 | 75.0000 | 0.4312 | 0.1155 |
| 5 | 0.4464 | 102.1900 | 0.4363 | -0.0943 |
| 6 | 0.4464 | 130.0000 | 0.3420 | -0.2869 |
| 7 | 0.4464 | 155.0000 | 0.1887 | -0.4046 |
| 8 | 0.4464 | 180.0000 | 0.0000 | -0.4464 |

BODY STATION X = 4.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.5943 | 0.0000 | 0.0000 | 0.5943 |
| 2 | 0.5943 | 25.0000 | 0.2512 | 0.5386 |
| 3 | 0.5943 | 50.0000 | 0.4553 | 0.3820 |
| 4 | 0.5943 | 75.0000 | 0.5740 | 0.1538 |
| 5 | 0.5943 | 102.1900 | 0.5809 | -0.1255 |
| 6 | 0.5943 | 130.0000 | 0.4553 | -0.3820 |
| 7 | 0.5943 | 155.0000 | 0.2512 | -0.5386 |
| 8 | 0.5943 | 180.0000 | 0.0000 | -0.5943 |

BODY STATION X = 6.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.7234 | 0.0000 | 0.0000 | 0.7234 |
| 2 | 0.7234 | 25.0000 | 0.3057 | 0.6556 |
| 3 | 0.7234 | 50.0000 | 0.5542 | 0.4650 |
| 4 | 0.7234 | 75.0000 | 0.6988 | 0.1872 |
| 5 | 0.7234 | 102.1900 | 0.7071 | -0.1527 |
| 6 | 0.7234 | 130.0000 | 0.5542 | -0.4650 |
| 7 | 0.7234 | 155.0000 | 0.3057 | -0.6556 |
| 8 | 0.7234 | 180.0000 | 0.0000 | -0.7234 |

BODY STATION X = 7.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.8370 | 0.0000 | 0.0000 | 0.8370 |
| 2 | 0.8370 | 25.0000 | 0.3537 | 0.7586 |
| 3 | 0.8370 | 50.0000 | 0.6412 | 0.5380 |
| 4 | 0.8370 | 75.0000 | 0.8085 | 0.2166 |
| 5 | 0.8370 | 102.1900 | 0.8181 | -0.1767 |
| 6 | 0.8370 | 130.0000 | 0.6412 | -0.5380 |
| 7 | 0.8370 | 155.0000 | 0.3537 | -0.7586 |
| 8 | 0.8370 | 180.0000 | 0.0000 | -0.8370 |

BODY STATION X = 9.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 0.9364 | 0.0000 | 0.0000 | 0.9364 |
| 2 | 0.9364 | 25.0000 | 0.3957 | 0.8487 |
| 3 | 0.9364 | 50.0000 | 0.7173 | 0.6019 |
| 4 | 0.9364 | 75.0000 | 0.9045 | 0.2424 |
| 5 | 0.9364 | 102.1900 | 0.9153 | -0.1977 |
| 6 | 0.9364 | 130.0000 | 0.7173 | -0.6019 |
| 7 | 0.9364 | 155.0000 | 0.3957 | -0.8487 |
| 8 | 0.9364 | 180.0000 | 0.0000 | -0.9364 |

BODY STATION X = 10.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.0223 | 0.0000 | 0.0000 | 1.0223 |
| 2 | 1.0223 | 25.0000 | 0.4320 | 0.9265 |
| 3 | 1.0223 | 50.0000 | 0.7831 | 0.6571 |
| 4 | 1.0223 | 75.0000 | 0.9875 | 0.2646 |
| 5 | 1.0223 | 102.1900 | 0.9992 | -0.2159 |
| 6 | 1.0223 | 130.0000 | 0.7831 | -0.6571 |
| 7 | 1.0223 | 155.0000 | 0.4320 | -0.9265 |
| 8 | 1.0223 | 180.0000 | 0.0000 | -1.0223 |

BODY STATION X = 12.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.0936 | 0.0000 | 0.0000 | 1.0936 |
| 2 | 1.0936 | 25.0000 | 0.4622 | 0.9911 |
| 3 | 1.0936 | 50.0000 | 0.8377 | 0.7030 |
| 4 | 1.0936 | 75.0000 | 1.0563 | 0.2830 |
| 5 | 1.0936 | 102.1900 | 1.0689 | -0.2309 |
| 6 | 1.0936 | 130.0000 | 0.8377 | -0.7030 |
| 7 | 1.0936 | 155.0000 | 0.4622 | -0.9911 |
| 8 | 1.0936 | 180.0000 | 0.0000 | -1.0936 |

BODY STATION X = 13.5000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1479 | 0.0000 | 0.0000 | 1.1479 |
| 2 | 1.1479 | 25.0000 | 0.4851 | 1.0404 |
| 3 | 1.1479 | 50.0000 | 0.8793 | 0.7379 |
| 4 | 1.1479 | 75.0000 | 1.1088 | 0.2971 |
| 5 | 1.1479 | 102.1900 | 1.1220 | -0.2424 |
| 6 | 1.1479 | 130.0000 | 0.8793 | -0.7379 |
| 7 | 1.1479 | 155.0000 | 0.4851 | -1.0404 |
| 8 | 1.1479 | 180.0000 | 0.0000 | -1.1479 |

BODY STATION X = 15.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 15.1000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 17.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 20.0000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 22.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 25.0000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 27.5000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 29.0000 SCODE = 0.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1840 | 0.0000 | 0.0000 | 1.1840 |
| 2 | 1.1840 | 25.0000 | 0.5004 | 1.0731 |
| 3 | 1.1840 | 50.0000 | 0.9070 | 0.7611 |
| 4 | 1.1840 | 75.0000 | 1.1437 | 0.3064 |
| 5 | 1.1840 | 102.1900 | 1.1573 | -0.2500 |
| 6 | 1.1840 | 130.0000 | 0.9070 | -0.7611 |
| 7 | 1.1840 | 155.0000 | 0.5004 | -1.0731 |
| 8 | 1.1840 | 180.0000 | 0.0000 | -1.1840 |

BODY STATION X = 31.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1740 | 0.0000 | 0.0000 | 1.1740 |
| 2 | 1.1740 | 25.0000 | 0.4962 | 1.0640 |
| 3 | 1.1740 | 50.0000 | 0.8993 | 0.7546 |
| 4 | 1.1740 | 75.0000 | 1.1340 | 0.3039 |
| 5 | 1.1740 | 102.1900 | 1.1475 | -0.2479 |
| 6 | 1.1740 | 130.0000 | 0.8993 | -0.7546 |
| 7 | 1.1740 | 155.0000 | 0.4962 | -1.0640 |
| 8 | 1.1740 | 180.0000 | 0.0000 | -1.1740 |

BODY STATION X = 32.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1540 | 0.0000 | 0.0000 | 1.1540 |
| 2 | 1.1540 | 25.0000 | 0.4877 | 1.0459 |
| 3 | 1.1540 | 50.0000 | 0.8840 | 0.7418 |
| 4 | 1.1540 | 75.0000 | 1.1147 | 0.2987 |
| 5 | 1.1540 | 102.1900 | 1.1280 | -0.2437 |
| 6 | 1.1540 | 130.0000 | 0.8840 | -0.7418 |
| 7 | 1.1540 | 155.0000 | 0.4877 | -1.0459 |
| 8 | 1.1540 | 180.0000 | 0.0000 | -1.1540 |

BODY STATION X = 33.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.1240 | 0.0000 | 0.0000 | 1.1240 |
| 2 | 1.1240 | 25.0000 | 0.4750 | 1.0187 |
| 3 | 1.1240 | 50.0000 | 0.8610 | 0.7225 |
| 4 | 1.1240 | 75.0000 | 1.0857 | 0.2909 |
| 5 | 1.1240 | 102.1900 | 1.0987 | -0.2373 |
| 6 | 1.1240 | 130.0000 | 0.8610 | -0.7225 |
| 7 | 1.1240 | 155.0000 | 0.4750 | -1.0187 |
| 8 | 1.1240 | 180.0000 | 0.0000 | -1.1240 |

BODY STATION X = 34.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.0840 | 0.0000 | 0.0000 | 1.0840 |
| 2 | 1.0840 | 25.0000 | 0.4581 | 0.9824 |
| 3 | 1.0840 | 50.0000 | 0.8304 | 0.6968 |
| 4 | 1.0840 | 75.0000 | 1.0471 | 0.2806 |
| 5 | 1.0840 | 102.1900 | 1.0596 | -0.2289 |
| 6 | 1.0840 | 130.0000 | 0.8304 | -0.6968 |
| 7 | 1.0840 | 155.0000 | 0.4581 | -0.9824 |
| 8 | 1.0840 | 180.0000 | 0.0000 | -1.0840 |

BODY STATION X = 35.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.0340 | 0.0000 | 0.0000 | 1.0340 |
| 2 | 1.0340 | 25.0000 | 0.4370 | 0.9371 |
| 3 | 1.0340 | 50.0000 | 0.7921 | 0.6646 |
| 4 | 1.0340 | 75.0000 | 0.9988 | 0.2676 |
| 5 | 1.0340 | 102.1900 | 1.0107 | -0.2183 |
| 6 | 1.0340 | 130.0000 | 0.7921 | -0.6646 |
| 7 | 1.0340 | 155.0000 | 0.4370 | -0.9371 |
| 8 | 1.0340 | 180.0000 | 0.0000 | -1.0340 |

BODY STATION X = 36.0000 SCODE = 2.
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| MERIDIAN NO. | RHO | THETA | Y | Z |
|---|---|---|---|---|
| 1 | 1.0000 | 0.0000 | 0.0000 | 1.0000 |
| 2 | 1.0000 | 25.0000 | 0.4226 | 0.9063 |
| 3 | 1.0000 | 50.0000 | 0.7660 | 0.6428 |
| 4 | 1.0000 | 75.0000 | 0.9659 | 0.2588 |
| 5 | 1.0000 | 102.1900 | 0.9775 | -0.2112 |
| 6 | 1.0000 | 130.0000 | 0.7660 | -0.6428 |
| 7 | 1.0000 | 155.0000 | 0.4226 | -0.9063 |
| 8 | 1.0000 | 180.0000 | 0.0000 | -1.0000 |

BODY MERIDIAN LINE 1

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.0000 | 0.2700 | * | 0.2700 | 0.0000 |
| 3 | 3.0000 | 0.0000 | 0.4464 | * | 0.4464 | 0.0000 |
| 4 | 4.5000 | 0.0000 | 0.5943 | * | 0.5943 | 0.0000 |
| 5 | 6.0000 | 0.0000 | 0.7234 | * | 0.7234 | 0.0000 |
| 6 | 7.5000 | 0.0000 | 0.8370 | * | 0.8370 | 0.0000 |
| 7 | 9.0000 | 0.0000 | 0.9364 | * | 0.9364 | 0.0000 |
| 8 | 10.5000 | 0.0000 | 1.0223 | * | 1.0223 | 0.0000 |
| 9 | 12.0000 | 0.0000 | 1.0936 | * | 1.0936 | 0.0000 |
| 10 | 13.5000 | 0.0000 | 1.1479 | * | 1.1479 | 0.0000 |
| 11 | 15.0000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 12 | 15.1000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 13 | 17.5000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 14 | 20.0000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 15 | 22.5000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 16 | 25.0000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 17 | 27.5000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 18 | 29.9000 | 0.0000 | 1.1840 | * | 1.1840 | 0.0000 |
| 19 | 31.0000 | 0.0000 | 1.1740 | * | 1.1740 | 0.0000 |
| 20 | 32.0000 | 0.0000 | 1.1540 | * | 1.1540 | 0.0000 |
| 21 | 33.0000 | 0.0000 | 1.1240 | * | 1.1240 | 0.0000 |
| 22 | 34.0000 | 0.0000 | 1.0840 | * | 1.0840 | 0.0000 |
| 23 | 35.0000 | 0.0000 | 1.0340 | * | 1.0340 | 0.0000 |
| 24 | 36.0000 | 0.0000 | 1.0000 | * | 1.0000 | 0.0000 |

BODY MERIDIAN LINE 2

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.1141 | 0.2447 | * | 0.2700 | 25.0000 |
| 3 | 3.0000 | 0.1887 | 0.4046 | * | 0.4464 | 25.0000 |
| 4 | 4.5000 | 0.2512 | 0.5386 | * | 0.5943 | 25.0000 |
| 5 | 6.0000 | 0.3057 | 0.6556 | * | 0.7234 | 25.0000 |
| 6 | 7.5000 | 0.3537 | 0.7586 | * | 0.8370 | 25.0000 |
| 7 | 9.0000 | 0.3957 | 0.8487 | * | 0.9364 | 25.0000 |
| 8 | 10.5000 | 0.4320 | 0.9265 | * | 1.0223 | 25.0000 |
| 9 | 12.0000 | 0.4622 | 0.9911 | * | 1.0936 | 25.0000 |
| 10 | 13.5000 | 0.4851 | 1.0404 | * | 1.1479 | 25.0000 |
| 11 | 15.0000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 12 | 15.1000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 13 | 17.5000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 14 | 20.0000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 15 | 22.5000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 16 | 25.0000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 17 | 27.5000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 18 | 29.9000 | 0.5004 | 1.0731 | * | 1.1840 | 25.0000 |
| 19 | 31.0000 | 0.4962 | 1.0640 | * | 1.1740 | 25.0000 |
| 20 | 32.0000 | 0.4877 | 1.0459 | * | 1.1540 | 25.0000 |
| 21 | 33.0000 | 0.4750 | 1.0187 | * | 1.1240 | 25.0000 |
| 22 | 34.0000 | 0.4581 | 0.9824 | * | 1.0840 | 25.0000 |
| 23 | 35.0000 | 0.4370 | 0.9371 | * | 1.0340 | 25.0000 |
| 24 | 36.0000 | 0.4226 | 0.9063 | * | 1.0000 | 25.0000 |

BODY MERIDIAN LINE 3

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2068 | 0.1736 | * | 0.2700 | 50.0000 |
| 3 | 3.0000 | 0.3420 | 0.2869 | * | 0.4464 | 50.0000 |
| 4 | 4.5000 | 0.4553 | 0.3820 | * | 0.5943 | 50.0000 |
| 5 | 6.0000 | 0.5542 | 0.4650 | * | 0.7234 | 50.0000 |
| 6 | 7.5000 | 0.6412 | 0.5380 | * | 0.8370 | 50.0000 |
| 7 | 9.0000 | 0.7173 | 0.6019 | * | 0.9364 | 50.0000 |
| 8 | 10.5000 | 0.7831 | 0.6571 | * | 1.0223 | 50.0000 |
| 9 | 12.0000 | 0.8377 | 0.7030 | * | 1.0936 | 50.0000 |
| 10 | 13.5000 | 0.8793 | 0.7379 | * | 1.1479 | 50.0000 |
| 11 | 15.0000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 12 | 15.1000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 13 | 17.5000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 14 | 20.0000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 15 | 22.5000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 16 | 25.0000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 17 | 27.5000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 18 | 29.9000 | 0.9070 | 0.7611 | * | 1.1840 | 50.0000 |
| 19 | 31.0000 | 0.8993 | 0.7546 | * | 1.1740 | 50.0000 |
| 20 | 32.0000 | 0.8840 | 0.7418 | * | 1.1540 | 50.0000 |
| 21 | 33.0000 | 0.8610 | 0.7225 | * | 1.1240 | 50.0000 |
| 22 | 34.0000 | 0.8304 | 0.6968 | * | 1.0840 | 50.0000 |
| 23 | 35.0000 | 0.7921 | 0.6646 | * | 1.0340 | 50.0000 |
| 24 | 36.0000 | 0.7660 | 0.6428 | * | 1.0000 | 50.0000 |

BODY MERIDIAN LINE 4

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2608 | 0.0699 | * | 0.2700 | 75.0000 |
| 3 | 3.0000 | 0.4312 | 0.1155 | * | 0.4464 | 75.0000 |
| 4 | 4.5000 | 0.5740 | 0.1538 | * | 0.5943 | 75.0000 |
| 5 | 6.0000 | 0.6988 | 0.1872 | * | 0.7234 | 75.0000 |
| 6 | 7.5000 | 0.8085 | 0.2166 | * | 0.8370 | 75.0000 |
| 7 | 9.0000 | 0.9045 | 0.2424 | * | 0.9364 | 75.0000 |
| 8 | 10.5000 | 0.9875 | 0.2646 | * | 1.0223 | 75.0000 |
| 9 | 12.0000 | 1.0563 | 0.2830 | * | 1.0936 | 75.0000 |
| 10 | 13.5000 | 1.1088 | 0.2971 | * | 1.1479 | 75.0000 |
| 11 | 15.0000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 12 | 15.1000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 13 | 17.5000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 14 | 20.0000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 15 | 22.5000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 16 | 25.0000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 17 | 27.5000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 18 | 29.9000 | 1.1437 | 0.3064 | * | 1.1840 | 75.0000 |
| 19 | 31.0000 | 1.1340 | 0.3039 | * | 1.1740 | 75.0000 |
| 20 | 32.0000 | 1.1147 | 0.2987 | * | 1.1540 | 75.0000 |
| 21 | 33.0000 | 1.0857 | 0.2909 | * | 1.1240 | 75.0000 |
| 22 | 34.0000 | 1.0471 | 0.2806 | * | 1.0840 | 75.0000 |
| 23 | 35.0000 | 0.9988 | 0.2676 | * | 1.0340 | 75.0000 |
| 24 | 36.0000 | 0.9659 | 0.2588 | * | 1.0000 | 75.0000 |

TR-805/EQUIVALENT/BODY

MAY 20, 1967

BODY MERIDIAN LINE 5

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | -0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2639 | -0.0570 | * | 0.2700 | 102.1900 |
| 3 | 3.0000 | 0.4363 | -0.0943 | * | 0.4464 | 102.1900 |
| 4 | 4.5000 | 0.5809 | -0.1255 | * | 0.5943 | 102.1900 |
| 5 | 6.0000 | 0.7071 | -0.1527 | * | 0.7234 | 102.1900 |
| 6 | 7.5000 | 0.8181 | -0.1767 | * | 0.8370 | 102.1900 |
| 7 | 9.0000 | 0.9153 | -0.1977 | * | 0.9364 | 102.1900 |
| 8 | 10.5000 | 0.9992 | -0.2159 | * | 1.0223 | 102.1900 |
| 9 | 12.0000 | 1.0689 | -0.2309 | * | 1.0936 | 102.1900 |
| 10 | 13.5000 | 1.1220 | -0.2424 | * | 1.1479 | 102.1900 |
| 11 | 15.0000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 12 | 15.1000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 13 | 17.5000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 14 | 20.0000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 15 | 22.5000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 16 | 25.0000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 17 | 27.5000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 18 | 29.9000 | 1.1573 | -0.2500 | * | 1.1840 | 102.1900 |
| 19 | 31.0000 | 1.1475 | -0.2479 | * | 1.1740 | 102.1900 |
| 20 | 32.0000 | 1.1280 | -0.2437 | * | 1.1540 | 102.1900 |
| 21 | 33.0000 | 1.0987 | -0.2373 | * | 1.1240 | 102.1900 |
| 22 | 34.0000 | 1.0596 | -0.2289 | * | 1.0840 | 102.1900 |
| 23 | 35.0000 | 1.0107 | -0.2183 | * | 1.0340 | 102.1900 |
| 24 | 36.0000 | 0.9775 | -0.2112 | * | 1.0000 | 102.1900 |

BODY MERIDIAN LINE 6

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | -0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2068 | -0.1736 | * | 0.2700 | 130.0000 |
| 3 | 3.0000 | 0.3420 | -0.2869 | * | 0.4464 | 130.0000 |
| 4 | 4.5000 | 0.4553 | -0.3820 | * | 0.5943 | 130.0000 |
| 5 | 6.0000 | 0.5542 | -0.4650 | * | 0.7234 | 130.0000 |
| 6 | 7.5000 | 0.6412 | -0.5380 | * | 0.8370 | 130.0000 |
| 7 | 9.0000 | 0.7173 | -0.6019 | * | 0.9364 | 130.0000 |
| 8 | 10.5000 | 0.7831 | -0.6571 | * | 1.0223 | 130.0000 |
| 9 | 12.0000 | 0.8377 | -0.7030 | * | 1.0936 | 130.0000 |
| 10 | 13.5000 | 0.8793 | -0.7379 | * | 1.1479 | 130.0000 |
| 11 | 15.0000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 12 | 15.1000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 13 | 17.5000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 14 | 20.0000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 15 | 22.5000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 16 | 25.0000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 17 | 27.5000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 18 | 29.9000 | 0.9070 | -0.7611 | * | 1.1840 | 130.0000 |
| 19 | 31.0000 | 0.8993 | -0.7546 | * | 1.1740 | 130.0000 |
| 20 | 32.0000 | 0.8840 | -0.7418 | * | 1.1540 | 130.0000 |
| 21 | 33.0000 | 0.8610 | -0.7225 | * | 1.1240 | 130.0000 |
| 22 | 34.0000 | 0.8304 | -0.6968 | * | 1.0840 | 130.0000 |
| 23 | 35.0000 | 0.7921 | -0.6646 | * | 1.0340 | 130.0000 |
| 24 | 36.0000 | 0.7660 | -0.6428 | * | 1.0000 | 130.0000 |

BODY MERIDIAN LINE 7

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | -0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.1141 | -0.2447 | * | 0.2700 | 155.0000 |
| 3 | 3.0000 | 0.1887 | -0.4046 | * | 0.4464 | 155.0000 |
| 4 | 4.5000 | 0.2512 | -0.5386 | * | 0.5943 | 155.0000 |
| 5 | 6.0000 | 0.3057 | -0.6556 | * | 0.7234 | 155.0000 |
| 6 | 7.5000 | 0.3537 | -0.7586 | * | 0.8370 | 155.0000 |
| 7 | 9.0000 | 0.3957 | -0.8487 | * | 0.9364 | 155.0000 |
| 8 | 10.5000 | 0.4320 | -0.9265 | * | 1.0223 | 155.0000 |
| 9 | 12.0000 | 0.4622 | -0.9911 | * | 1.0936 | 155.0000 |
| 10 | 13.5000 | 0.4851 | -1.0404 | * | 1.1479 | 155.0000 |
| 11 | 15.0000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 12 | 15.1000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 13 | 17.5000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 14 | 20.0000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 15 | 22.5000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 16 | 25.0000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 17 | 27.5000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 18 | 29.9000 | 0.5004 | -1.0731 | * | 1.1840 | 155.0000 |
| 19 | 31.0000 | 0.4962 | -1.0640 | * | 1.1740 | 155.0000 |
| 20 | 32.0000 | 0.4877 | -1.0459 | * | 1.1540 | 155.0000 |
| 21 | 33.0000 | 0.4750 | -1.0187 | * | 1.1240 | 155.0000 |
| 22 | 34.0000 | 0.4581 | -0.9824 | * | 1.0840 | 155.0000 |
| 23 | 35.0000 | 0.4370 | -0.9371 | * | 1.0340 | 155.0000 |
| 24 | 36.0000 | 0.4226 | -0.9063 | * | 1.0000 | 155.0000 |

BODY MERIDIAN LINE 8

| | X | Y | Z | | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | -0.0000 | * | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.0000 | -0.2700 | * | 0.2700 | 180.0000 |
| 3 | 3.0000 | 0.0000 | -0.4464 | * | 0.4464 | 180.0000 |
| 4 | 4.5000 | 0.0000 | -0.5943 | * | 0.5943 | 180.0000 |
| 5 | 6.0000 | 0.0000 | -0.7234 | * | 0.7234 | 180.0000 |
| 6 | 7.5000 | 0.0000 | -0.8370 | * | 0.8370 | 180.0000 |
| 7 | 9.0000 | 0.0000 | -0.9364 | * | 0.9364 | 180.0000 |
| 8 | 10.5000 | 0.0000 | -1.0223 | * | 1.0223 | 180.0000 |
| 9 | 12.0000 | 0.0000 | -1.0936 | * | 1.0936 | 180.0000 |
| 10 | 13.5000 | 0.0000 | -1.1479 | * | 1.1479 | 180.0000 |
| 11 | 15.0000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 12 | 15.1000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 13 | 17.5000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 14 | 20.0000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 15 | 22.5000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 16 | 25.0000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 17 | 27.5000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 18 | 29.9000 | 0.0000 | -1.1840 | * | 1.1840 | 180.0000 |
| 19 | 31.0000 | 0.0000 | -1.1740 | * | 1.1740 | 180.0000 |
| 20 | 32.0000 | 0.0000 | -1.1540 | * | 1.1540 | 180.0000 |
| 21 | 33.0000 | 0.0000 | -1.1240 | * | 1.1240 | 180.0000 |
| 22 | 34.0000 | 0.0000 | -1.0840 | * | 1.0840 | 180.0000 |
| 23 | 35.0000 | 0.0000 | -1.0340 | * | 1.0340 | 180.0000 |
| 24 | 36.0000 | 0.0000 | -1.0000 | * | 1.0000 | 180.0000 |

```
                                                                      100
WING
                                                     MAY 20, 1967
WING DEFINITION

TR-805 WING(TWP=-.25)

THE PLANFORM HAS  2 POINTS ON THE LEADING EDGE,
                  3 POINTS ON THE TRAILING EDGE.
  4 AIRFOILS ARE GIVEN.  11 PERCENT LINES ARE TO BE GENERATED.

PCODE = 1., ACODE = 1., EPS =  0.010000, CHD =-0.0000
UPPER WING SURFACE IS DEFINED

PLANFORM LEADING  EDGE POINTS (GIVEN) ARE (X,Y) =
   10.6700    0.0000        40.2700   10.7740

PLANFORM TRAILING EDGE POINTS (GIVEN) ARE (X,Y) =
   19.8800    0.0000        43.5180    8.6030        40.2700   10.7740


AIRFOIL NO.  1
AFK =  3.,     AFANG =   0.0000,     YL =     0.0000,     YT =     0.0000
AFNU = 2.,      AFNL =-0.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
           10.6700    0.0000                19.8800    0.0000
CHORD LENGTH =     9.2100

THE GIVEN   2 UPPER AIRFOIL POINTS ARE
    XWING                  YWING
     0.0000                -0.2500
     9.2100                -0.2500

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)          Z
     0.0000                -0.2500
     9.2100                -0.2500


AIRFOIL NO.  2
AFK =  1.,     AFANG =   0.0000,     YL =     5.0000,     YT =     0.0000
AFNU = 0.,      AFNL =-0.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
           24.4068    5.0000                33.6182    5.0000
CHORD LENGTH =     9.2115

THE GIVEN UPPER AIRFOIL IS THE SAME AS PRECEDING ONE

AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
    XWING (SCALED)          Z
     0.0000                -0.2500
     9.2115                -0.2500
```

```
WING DEFINITION                                                    MAY 20, 1967

TR-805 WING(TWP=-.25)

 AIRFOIL NO.   3
 AFK =  3.,     AFANG =  -0.0000,     YL =     8.6030,     YT =     8.6030
 AFNU = 2.,      AFNL =-0.
 LEADING AND TRAILING EDGE POINTS (X,Y) ARE
             34.3055    8.6030                  43.5180    8.6030
 CHORD LENGTH =     9.2125

 THE GIVEN    2 UPPER AIRFOIL POINTS ARE
     XWING                      YWING
      0.0000                    -0.2500
      9.2100                    -0.2500

 AIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE
     XWING (SCALED)              Z
      0.0000                    -0.2501
      9.2125                    -0.2501


AIRFOIL NO.  4
AFK =  3.,     AFANG =  -0.0000,     YL =   10.7740,     YT =   10.7740
AFNU = 2.,      AFNL =-0.
LEADING AND TRAILING EDGE POINTS (X,Y) ARE
            40.2700   10.7740                 40.2700   10.7740
CHORD LENGTH =     0.0000

THE GIVEN    2 UPPER AIRFOIL POINTS ARE
    XWING                      YWING
     0.0000                    -0.2500
     0.0000                    -0.2500

THE AIRFOIL IS A POINT
```

```
PLANFORM LEADING  EDGE POINTS (USED)  ARE (X,Y) =
   10.6700    0.0000        24.4068    5.0000        34.3055    8.6030
   40.2700   10.7740

PLANFORM TRAILING EDGE POINTS (USED)  ARE (X,Y) =
   19.8800    0.0000        33.6182    5.0000        43.5180    8.6030
   40.2700   10.7740

THE FOLLOWING PERCENT CHORD LINES ARE TO BE CONSTRUCTED
    0.0000   10.0000   20.0000   30.0000   40.0000   50.0000
   60.0000   70.0000   80.0000   90.0000  100.0000
```

TR-805 WING(TWP=-.25)
POINTS ON UPPER WING SURFACE PERCENT CHORD LINES MAY 20, 1967

WING PERCENT CHORD LINE 1, 0.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 10.6700 | 0.0000 | -0.2500 |
| 2 | 24.4068 | 5.0000 | -0.2500 |
| 3 | 34.3055 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 2, 10.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 11.5910 | 0.0000 | -0.2500 |
| 2 | 25.3279 | 5.0000 | -0.2500 |
| 3 | 35.2267 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 3, 20.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 12.5120 | 0.0000 | -0.2500 |
| 2 | 26.2491 | 5.0000 | -0.2500 |
| 3 | 36.1480 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 4, 30.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 13.4330 | 0.0000 | -0.2500 |
| 2 | 27.1702 | 5.0000 | -0.2500 |
| 3 | 37.0692 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 5, 40.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 14.3540 | 0.0000 | -0.2500 |
| 2 | 28.0914 | 5.0000 | -0.2500 |
| 3 | 37.9905 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 6, 50.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 15.2750 | 0.0000 | -0.2500 |
| 2 | 29.0125 | 5.0000 | -0.2500 |
| 3 | 38.9117 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 7, 60.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 16.1960 | 0.0000 | -0.2500 |
| 2 | 29.9335 | 5.0000 | -0.2500 |
| 3 | 39.8330 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

TR-805 WING(TWP=-.25)
POINTS ON UPPER WING SURFACE PERCENT CHORD LINES      MAY 20, 1967

WING PERCENT CHORD LINE 8, 70.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 17.1170 | 0.0000 | -0.2500 |
| 2 | 30.8548 | 5.0000 | -0.2500 |
| 3 | 40.7542 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 9, 80.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 18.0380 | 0.0000 | -0.2500 |
| 2 | 31.7759 | 5.0000 | -0.2500 |
| 3 | 41.6755 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 10, 90.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 18.9590 | 0.0000 | -0.2500 |
| 2 | 32.6971 | 5.0000 | -0.2500 |
| 3 | 42.5967 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 11, 100.0000 PERCENT

| AF NO. | X | Y | Z |
|---|---|---|---|
| 1 | 19.8800 | 0.0000 | -0.2500 |
| 2 | 33.6182 | 5.0000 | -0.2500 |
| 3 | 43.5180 | 8.6030 | -0.2501 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

```
FIND THE INTERSECTION OF EACH WING PERCENT LINE WITH THE BODY SURFACE.

          CODE =  2

          TOLERANCE = 0.0001

LET P BE A POINT WHICH LIES ON THE BODY SURFACE
AND Q A POINT ON A LINE SEGMENT CONNECTING TWO ADJACENT
WING PERCENT LINE POINTS, SUCH THAT P AND Q HAVE THE SAME X-COORDINATE.

THE PROGRAM ITERATES UNTIL P AND Q HAVE THE SAME
(OR NEARLY THE SAME) Y AND Z COORDINATES.
THEN P IS THE INTERSECTION POINT.

IF P AND Q DIFFER BY MORE THAN THE GIVEN TOLERANCE,
A WARNING MESSAGE IS WRITTEN.

THE SHAPE OF THE BODY SURFACE BETWEEN MERIDIAN LINES (IN ANY SECTION
PARALLEL TO THE YZ PLANE) IS DETERMINED BY THE GIVEN CODE.
    IF CODE = 1, LINEAR INTERPOLATION IS USED BETWEEN MERIDIAN LINES.
    IF CODE = 2, BIQUADRATIC INTERPOLATION IS USED.
```

TR-805/EQUIVALENT/BODY
TR-805 WING(TWP=-.25)

MAY 20, 1967

INTERSECTIONS OF UPPER WING PERCENT CHORD LINES WITH BODY SURFACE
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( -0.0000, -0.0000)

| NO. | PERCENT | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 13.7701 | 1.1284 | -0.2500 | 1.1558 | 102.4926 |
| 2 | 10.0000 | 14.7548 | 1.1516 | -0.2500 | 1.1784 | 102.2492 |
| 3 | 20.0000 | 15.6916 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 4 | 30.0000 | 16.6126 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 5 | 40.0000 | 17.5337 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 6 | 50.0000 | 18.4547 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 7 | 60.0000 | 19.3757 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 8 | 70.0000 | 20.2968 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 9 | 80.0000 | 21.2178 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 10 | 90.0000 | 22.1388 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 11 | 100.0000 | 23.0599 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |

MAY 20, 1967

PROGRAM ENTERED LINK TO TRANSFORM BODY AND/OR WING BEFORE PANELLING

THE ORIGIN OF THE NEW BODY COORDINATE SYSTEM IS AT
(X,Y,Z) = ( 0.0000, 0.0000, 0.0000)

A UNIT VECTOR ALONG THE X-PRIME AXIS HAS COMPONENTS
(X,Y,Z) = ( 1.0000, 0.0000, 0.0000)

| NO. | ORIGINAL BODY STATION | CIRCULARIZED BODY STATION | BODY RADIUS | Z-BODY CENTROID |
|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 1.5000 | 0.2700 | 0.0000 |
| 3 | 3.0000 | 3.0000 | 0.4464 | 0.0000 |
| 4 | 4.5000 | 4.5000 | 0.5943 | 0.0000 |
| 5 | 6.0000 | 6.0000 | 0.7234 | 0.0000 |
| 6 | 7.5000 | 7.5000 | 0.8370 | 0.0000 |
| 7 | 9.0000 | 9.0000 | 0.9364 | 0.0000 |
| 8 | 10.5000 | 10.5000 | 1.0223 | 0.0000 |
| 9 | 12.0000 | 12.0000 | 1.0936 | 0.0000 |
| 10 | 13.5000 | 13.5000 | 1.1479 | 0.0000 |
| 11 | 15.0000 | 15.0000 | 1.1840 | 0.0000 |
| 12 | 15.1000 | 15.1000 | 1.1840 | 0.0000 |
| 13 | 17.5000 | 17.5000 | 1.1840 | 0.0000 |
| 14 | 20.0000 | 20.0000 | 1.1840 | 0.0000 |
| 15 | 22.5000 | 22.5000 | 1.1840 | 0.0000 |
| 16 | 25.0000 | 25.0000 | 1.1840 | 0.0000 |
| 17 | 27.5000 | 27.5000 | 1.1840 | 0.0000 |
| 18 | 29.9000 | 29.9000 | 1.1840 | 0.0000 |
| 19 | 31.0000 | 31.0000 | 1.1740 | 0.0000 |
| 20 | 32.0000 | 32.0000 | 1.1540 | 0.0000 |
| 21 | 33.0000 | 33.0000 | 1.1240 | 0.0000 |
| 22 | 34.0000 | 34.0000 | 1.0840 | 0.0000 |
| 23 | 35.0000 | 35.0000 | 1.0340 | 0.0000 |
| 24 | 36.0000 | 36.0000 | 1.0000 | 0.0000 |

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

MAY 20, 1967

BODY MERIDIAN LINE 1

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | -0.0000 | 0.2700 | 0.2700 | 0.0000 |
| 3 | 3.0000 | -0.0000 | 0.4464 | 0.4464 | 0.0000 |
| 4 | 4.5000 | -0.0000 | 0.5943 | 0.5943 | 0.0000 |
| 5 | 6.0000 | -0.0000 | 0.7234 | 0.7234 | 0.0000 |
| 6 | 7.5000 | -0.0000 | 0.8370 | 0.8370 | 0.0000 |
| 7 | 9.0000 | -0.0000 | 0.9364 | 0.9364 | 0.0000 |
| 8 | 10.5000 | -0.0000 | 1.0223 | 1.0223 | 0.0000 |
| 9 | 12.0000 | -0.0000 | 1.0936 | 1.0936 | 0.0000 |
| 10 | 13.5000 | -0.0000 | 1.1479 | 1.1479 | 0.0000 |
| 11 | 15.0000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 12 | 15.1000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 13 | 17.5000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 14 | 20.0000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 15 | 22.5000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 16 | 25.0000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 17 | 27.5000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 18 | 29.9000 | -0.0000 | 1.1840 | 1.1840 | 0.0000 |
| 19 | 31.0000 | -0.0000 | 1.1740 | 1.1740 | 0.0000 |
| 20 | 32.0000 | -0.0000 | 1.1540 | 1.1540 | 0.0000 |
| 21 | 33.0000 | -0.0000 | 1.1240 | 1.1240 | 0.0000 |
| 22 | 34.0000 | -0.0000 | 1.0840 | 1.0840 | 0.0000 |
| 23 | 35.0000 | -0.0000 | 1.0340 | 1.0340 | 0.0000 |
| 24 | 36.0000 | -0.0000 | 1.0000 | 1.0000 | 0.0000 |

BODY MERIDIAN LINE 2

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.1141 | 0.2447 | 0.2700 | 25.0000 |
| 3 | 3.0000 | 0.1887 | 0.4046 | 0.4464 | 25.0000 |
| 4 | 4.5000 | 0.2512 | 0.5386 | 0.5943 | 25.0000 |
| 5 | 6.0000 | 0.3057 | 0.6556 | 0.7234 | 25.0000 |
| 6 | 7.5000 | 0.3537 | 0.7586 | 0.8370 | 25.0000 |
| 7 | 9.0000 | 0.3957 | 0.8487 | 0.9364 | 25.0000 |
| 8 | 10.5000 | 0.4320 | 0.9265 | 1.0223 | 25.0000 |
| 9 | 12.0000 | 0.4622 | 0.9911 | 1.0936 | 25.0000 |
| 10 | 13.5000 | 0.4851 | 1.0404 | 1.1479 | 25.0000 |
| 11 | 15.0000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 12 | 15.1000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 13 | 17.5000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 14 | 20.0000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 15 | 22.5000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 16 | 25.0000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 17 | 27.5000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 18 | 29.9000 | 0.5004 | 1.0731 | 1.1840 | 25.0000 |
| 19 | 31.0000 | 0.4962 | 1.0640 | 1.1740 | 25.0000 |
| 20 | 32.0000 | 0.4877 | 1.0459 | 1.1540 | 25.0000 |
| 21 | 33.0000 | 0.4750 | 1.0187 | 1.1240 | 25.0000 |
| 22 | 34.0000 | 0.4581 | 0.9824 | 1.0840 | 25.0000 |
| 23 | 35.0000 | 0.4370 | 0.9371 | 1.0340 | 25.0000 |
| 24 | 36.0000 | 0.4226 | 0.9063 | 1.0000 | 25.0000 |

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

MAY 20, 1967

BODY MERIDIAN LINE 3

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2068 | 0.1736 | 0.2700 | 50.0000 |
| 3 | 3.0000 | 0.3420 | 0.2869 | 0.4464 | 50.0000 |
| 4 | 4.5000 | 0.4553 | 0.3820 | 0.5943 | 50.0000 |
| 5 | 6.0000 | 0.5542 | 0.4650 | 0.7234 | 50.0000 |
| 6 | 7.5000 | 0.6412 | 0.5380 | 0.8370 | 50.0000 |
| 7 | 9.0000 | 0.7173 | 0.6019 | 0.9364 | 50.0000 |
| 8 | 10.5000 | 0.7831 | 0.6571 | 1.0223 | 50.0000 |
| 9 | 12.0000 | 0.8377 | 0.7030 | 1.0936 | 50.0000 |
| 10 | 13.5000 | 0.8793 | 0.7379 | 1.1479 | 50.0000 |
| 11 | 15.0000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 12 | 15.1000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 13 | 17.5000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 14 | 20.0000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 15 | 22.5000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 16 | 25.0000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 17 | 27.5000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 18 | 29.9000 | 0.9070 | 0.7611 | 1.1840 | 50.0000 |
| 19 | 31.0000 | 0.8993 | 0.7546 | 1.1740 | 50.0000 |
| 20 | 32.0000 | 0.8840 | 0.7418 | 1.1540 | 50.0000 |
| 21 | 33.0000 | 0.8610 | 0.7225 | 1.1240 | 50.0000 |
| 22 | 34.0000 | 0.8304 | 0.6968 | 1.0840 | 50.0000 |
| 23 | 35.0000 | 0.7921 | 0.6646 | 1.0340 | 50.0000 |
| 24 | 36.0000 | 0.7660 | 0.6428 | 1.0000 | 50.0000 |

BODY MERIDIAN LINE 4

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2608 | 0.0699 | 0.2700 | 75.0000 |
| 3 | 3.0000 | 0.4312 | 0.1155 | 0.4464 | 75.0000 |
| 4 | 4.5000 | 0.5740 | 0.1538 | 0.5943 | 75.0000 |
| 5 | 6.0000 | 0.6988 | 0.1872 | 0.7234 | 75.0000 |
| 6 | 7.5000 | 0.8085 | 0.2166 | 0.8370 | 75.0000 |
| 7 | 9.0000 | 0.9045 | 0.2424 | 0.9364 | 75.0000 |
| 8 | 10.5000 | 0.9875 | 0.2646 | 1.0223 | 75.0000 |
| 9 | 12.0000 | 1.0563 | 0.2830 | 1.0936 | 75.0000 |
| 10 | 13.5000 | 1.1088 | 0.2971 | 1.1479 | 75.0000 |
| 11 | 15.0000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 12 | 15.1000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 13 | 17.5000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 14 | 20.0000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 15 | 22.5000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 16 | 25.0000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 17 | 27.5000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 18 | 29.9000 | 1.1437 | 0.3064 | 1.1840 | 75.0000 |
| 19 | 31.0000 | 1.1340 | 0.3039 | 1.1740 | 75.0000 |
| 20 | 32.0000 | 1.1147 | 0.2987 | 1.1540 | 75.0000 |
| 21 | 33.0000 | 1.0857 | 0.2909 | 1.1240 | 75.0000 |
| 22 | 34.0000 | 1.0471 | 0.2806 | 1.0840 | 75.0000 |
| 23 | 35.0000 | 0.9988 | 0.2676 | 1.0340 | 75.0000 |
| 24 | 36.0000 | 0.9659 | 0.2588 | 1.0000 | 75.0000 |

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

MAY 20, 1967

BODY MERIDIAN LINE 5

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | -0.0000 | -0.0000 | -0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2639 | -0.0570 | 0.2700 | 102.1900 |
| 3 | 3.0000 | 0.4363 | -0.0943 | 0.4464 | 102.1900 |
| 4 | 4.5000 | 0.5809 | -0.1255 | 0.5943 | 102.1900 |
| 5 | 6.0000 | 0.7071 | -0.1527 | 0.7234 | 102.1900 |
| 6 | 7.5000 | 0.8181 | -0.1767 | 0.8370 | 102.1900 |
| 7 | 9.0000 | 0.9153 | -0.1977 | 0.9364 | 102.1900 |
| 8 | 10.5000 | 0.9992 | -0.2159 | 1.0223 | 102.1900 |
| 9 | 12.0000 | 1.0689 | -0.2309 | 1.0936 | 102.1900 |
| 10 | 13.5000 | 1.1220 | -0.2424 | 1.1479 | 102.1900 |
| 11 | 15.0000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 12 | 15.1000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 13 | 17.5000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 14 | 20.0000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 15 | 22.5000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 16 | 25.0000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 17 | 27.5000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 18 | 29.9000 | 1.1573 | -0.2500 | 1.1840 | 102.1900 |
| 19 | 31.0000 | 1.1475 | -0.2479 | 1.1740 | 102.1900 |
| 20 | 32.0000 | 1.1280 | -0.2437 | 1.1540 | 102.1900 |
| 21 | 33.0000 | 1.0987 | -0.2373 | 1.1240 | 102.1900 |
| 22 | 34.0000 | 1.0596 | -0.2289 | 1.0840 | 102.1900 |
| 23 | 35.0000 | 1.0107 | -0.2183 | 1.0340 | 102.1900 |
| 24 | 36.0000 | 0.9775 | -0.2112 | 1.0000 | 102.1900 |

BODY MERIDIAN LINE 6

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | -0.0000 | -0.0000 | -0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.2068 | -0.1736 | 0.2700 | 130.0000 |
| 3 | 3.0000 | 0.3420 | -0.2869 | 0.4464 | 130.0000 |
| 4 | 4.5000 | 0.4553 | -0.3820 | 0.5943 | 130.0000 |
| 5 | 6.0000 | 0.5542 | -0.4650 | 0.7234 | 130.0000 |
| 6 | 7.5000 | 0.6412 | -0.5380 | 0.8370 | 130.0000 |
| 7 | 9.0000 | 0.7173 | -0.6019 | 0.9364 | 130.0000 |
| 8 | 10.5000 | 0.7831 | -0.6571 | 1.0223 | 130.0000 |
| 9 | 12.0000 | 0.8377 | -0.7030 | 1.0936 | 130.0000 |
| 10 | 13.5000 | 0.8793 | -0.7379 | 1.1479 | 130.0000 |
| 11 | 15.0000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 12 | 15.1000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 13 | 17.5000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 14 | 20.0000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 15 | 22.5000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 16 | 25.0000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 17 | 27.5000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 18 | 29.9000 | 0.9070 | -0.7611 | 1.1840 | 130.0000 |
| 19 | 31.0000 | 0.8993 | -0.7546 | 1.1740 | 130.0000 |
| 20 | 32.0000 | 0.8840 | -0.7418 | 1.1540 | 130.0000 |
| 21 | 33.0000 | 0.8610 | -0.7225 | 1.1240 | 130.0000 |
| 22 | 34.0000 | 0.8304 | -0.6968 | 1.0840 | 130.0000 |
| 23 | 35.0000 | 0.7921 | -0.6646 | 1.0340 | 130.0000 |
| 24 | 36.0000 | 0.7660 | -0.6428 | 1.0000 | 130.0000 |

TRANSFORMED BODY (PRIMED-AXIS SYSTEM)

MAY 20, 1967

BODY MERIDIAN LINE 7

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | -0.0000 | -0.0000 | -0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.1141 | -0.2447 | 0.2700 | 155.0000 |
| 3 | 3.0000 | 0.1887 | -0.4046 | 0.4464 | 155.0000 |
| 4 | 4.5000 | 0.2512 | -0.5386 | 0.5943 | 155.0000 |
| 5 | 6.0000 | 0.3057 | -0.6556 | 0.7234 | 155.0000 |
| 6 | 7.5000 | 0.3537 | -0.7586 | 0.8370 | 155.0000 |
| 7 | 9.0000 | 0.3957 | -0.8487 | 0.9364 | 155.0000 |
| 8 | 10.5000 | 0.4320 | -0.9265 | 1.0223 | 155.0000 |
| 9 | 12.0000 | 0.4622 | -0.9911 | 1.0936 | 155.0000 |
| 10 | 13.5000 | 0.4851 | -1.0404 | 1.1479 | 155.0000 |
| 11 | 15.0000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 12 | 15.1000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 13 | 17.5000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 14 | 20.0000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 15 | 22.5000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 16 | 25.0000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 17 | 27.5000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 18 | 29.9000 | 0.5004 | -1.0731 | 1.1840 | 155.0000 |
| 19 | 31.0000 | 0.4962 | -1.0640 | 1.1740 | 155.0000 |
| 20 | 32.0000 | 0.4877 | -1.0459 | 1.1540 | 155.0000 |
| 21 | 33.0000 | 0.4750 | -1.0187 | 1.1240 | 155.0000 |
| 22 | 34.0000 | 0.4581 | -0.9824 | 1.0840 | 155.0000 |
| 23 | 35.0000 | 0.4370 | -0.9371 | 1.0340 | 155.0000 |
| 24 | 36.0000 | 0.4226 | -0.9063 | 1.0000 | 155.0000 |

BODY MERIDIAN LINE 8

| | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|
| 1 | -0.0000 | -0.0000 | -0.0000 | 0.0000 | 0.0000 |
| 2 | 1.5000 | 0.0000 | -0.2700 | 0.2700 | 180.0000 |
| 3 | 3.0000 | 0.0000 | -0.4464 | 0.4464 | 180.0000 |
| 4 | 4.5000 | 0.0000 | -0.5943 | 0.5943 | 180.0000 |
| 5 | 6.0000 | 0.0000 | -0.7234 | 0.7234 | 180.0000 |
| 6 | 7.5000 | 0.0000 | -0.8370 | 0.8370 | 180.0000 |
| 7 | 9.0000 | 0.0000 | -0.9364 | 0.9364 | 180.0000 |
| 8 | 10.5000 | 0.0000 | -1.0223 | 1.0223 | 180.0000 |
| 9 | 12.0000 | 0.0000 | -1.0936 | 1.0936 | 180.0000 |
| 10 | 13.5000 | 0.0000 | -1.1479 | 1.1479 | 180.0000 |
| 11 | 15.0000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 12 | 15.1000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 13 | 17.5000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 14 | 20.0000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 15 | 22.5000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 16 | 25.0000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 17 | 27.5000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 18 | 29.9000 | 0.0000 | -1.1840 | 1.1840 | 180.0000 |
| 19 | 31.0000 | 0.0000 | -1.1740 | 1.1740 | 180.0000 |
| 20 | 32.0000 | 0.0000 | -1.1540 | 1.1540 | 180.0000 |
| 21 | 33.0000 | 0.0000 | -1.1240 | 1.1240 | 180.0000 |
| 22 | 34.0000 | 0.0000 | -1.0840 | 1.0840 | 180.0000 |
| 23 | 35.0000 | 0.0000 | -1.0340 | 1.0340 | 180.0000 |
| 24 | 36.0000 | 0.0000 | -1.0000 | 1.0000 | 180.0000 |

POINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)

MAY 20, 1967

WING PERCENT CHORD LINE 1, 0.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 10.6700 | -0.0000 | -0.2500 |
| 2 | 24.4068 | 5.0000 | -0.2500 |
| 3 | 34.3055 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 2, 10.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 11.5910 | -0.0000 | -0.2500 |
| 2 | 25.3279 | 5.0000 | -0.2500 |
| 3 | 35.2267 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 3, 20.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 12.5120 | -0.0000 | -0.2500 |
| 2 | 26.2491 | 5.0000 | -0.2500 |
| 3 | 36.1480 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 4, 30.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 13.4330 | -0.0000 | -0.2500 |
| 2 | 27.1702 | 5.0000 | -0.2500 |
| 3 | 37.0692 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 5, 40.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 14.3540 | -0.0000 | -0.2500 |
| 2 | 28.0914 | 5.0000 | -0.2500 |
| 3 | 37.9905 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 6, 50.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 15.2750 | -0.0000 | -0.2500 |
| 2 | 29.0125 | 5.0000 | -0.2500 |
| 3 | 38.9117 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

POINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)

MAY 20, 1967

WING PERCENT CHORD LINE 7, 60.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 16.1960 | -0.0000 | -0.2500 |
| 2 | 29.9336 | 5.0000 | -0.2500 |
| 3 | 39.8330 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 8, 70.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 17.1170 | -0.0000 | -0.2500 |
| 2 | 30.8548 | 5.0000 | -0.2500 |
| 3 | 40.7542 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 9, 80.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 18.0380 | -0.0000 | -0.2500 |
| 2 | 31.7759 | 5.0000 | -0.2500 |
| 3 | 41.6755 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 10, 90.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 18.9590 | -0.0000 | -0.2500 |
| 2 | 32.6971 | 5.0000 | -0.2500 |
| 3 | 42.5967 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

WING PERCENT CHORD LINE 11, 100.0000

| | X | Y | Z |
|---|---|---|---|
| 1 | 19.8800 | -0.0000 | -0.2500 |
| 2 | 33.6182 | 5.0000 | -0.2500 |
| 3 | 43.5180 | 8.6030 | -0.2500 |
| 4 | 40.2700 | 10.7740 | -0.2500 |

MAY 20, 1967

INTERSECTIONS OF FLAT WING PERCENT CHORD LINES WITH BODY SURFACE
RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( 0.0000, 0.0000)

(PRIMED-AXIS SYSTEM)

| NO. | PERCENT | X | Y | Z | RHO | THETA |
|---|---|---|---|---|---|---|
| 1 | 0.0000 | 13.7610 | 1.1251 | -0.2500 | 1.1525 | 102.5285 |
| 2 | 10.0000 | 14.7529 | 1.1509 | -0.2500 | 1.1777 | 102.2562 |
| 3 | 20.0000 | 15.6916 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 4 | 30.0000 | 16.6126 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 5 | 40.0000 | 17.5337 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 6 | 50.0000 | 18.4547 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 7 | 60.0000 | 19.3757 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 8 | 70.0000 | 20.2968 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 9 | 80.0000 | 21.2178 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 10 | 90.0000 | 22.1388 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |
| 11 | 100.0000 | 23.0599 | 1.1573 | -0.2500 | 1.1840 | 102.1901 |

MAY 20, 1967

BODY RADIUS AND Z-COORDINATE OF BODY CENTROID VERSUS X-PRIME

| NO. | X-PRIME | RADIUS | Z-PRIME |
|---|---|---|---|
| 1 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.7243 | 0.1304 | 0.0000 |
| 3 | 1.4485 | 0.2607 | 0.0000 |
| 4 | 2.1728 | 0.3491 | 0.0000 |
| 5 | 2.8970 | 0.4343 | 0.0000 |
| 6 | 3.6213 | 0.5077 | 0.0000 |
| 7 | 4.3456 | 0.5791 | 0.0000 |
| 8 | 5.0698 | 0.6433 | 0.0000 |
| 9 | 5.7941 | 0.7057 | 0.0000 |
| 10 | 6.5184 | 0.7627 | 0.0000 |
| 11 | 7.2426 | 0.8175 | 0.0000 |
| 12 | 7.9669 | 0.8679 | 0.0000 |
| 13 | 8.6911 | 0.9159 | 0.0000 |
| 14 | 9.4154 | 0.9602 | 0.0000 |
| 15 | 10.1397 | 1.0017 | 0.0000 |
| 16 | 10.8639 | 1.0396 | 0.0000 |
| 17 | 11.5892 | 1.0740 | 0.0000 |
| 18 | 12.3124 | 1.1049 | 0.0000 |
| 19 | 13.0367 | 1.1311 | 0.0000 |
| 20 | 13.7610 | 1.1542 | 0.0000 |
| 21 | 14.5023 | 1.1720 | 0.0000 |
| 22 | 15.2436 | 1.1840 | 0.0000 |
| 23 | 15.9849 | 1.1840 | 0.0000 |
| 24 | 16.7252 | 1.1840 | 0.0000 |
| 25 | 17.4675 | 1.1840 | 0.0000 |
| 26 | 18.2088 | 1.1840 | 0.0000 |
| 27 | 18.9501 | 1.1840 | 0.0000 |
| 28 | 19.6914 | 1.1840 | 0.0000 |
| 29 | 20.4327 | 1.1840 | 0.0000 |
| 30 | 21.1740 | 1.1840 | 0.0000 |
| 31 | 21.9153 | 1.1840 | 0.0000 |
| 32 | 22.6566 | 1.1840 | 0.0000 |
| 33 | 23.3979 | 1.1840 | 0.0000 |
| 34 | 24.1392 | 1.1840 | 0.0000 |
| 35 | 24.8805 | 1.1840 | 0.0000 |
| 36 | 25.6218 | 1.1840 | 0.0000 |
| 37 | 26.3631 | 1.1840 | 0.0000 |
| 38 | 27.1044 | 1.1840 | 0.0000 |
| 39 | 27.8457 | 1.1840 | 0.0000 |
| 40 | 28.5870 | 1.1840 | 0.0000 |
| 41 | 29.3283 | 1.1840 | 0.0000 |
| 42 | 30.0696 | 1.1825 | 0.0000 |
| 43 | 30.8109 | 1.1757 | 0.0000 |
| 44 | 31.5522 | 1.1630 | 0.0000 |
| 45 | 32.2935 | 1.1452 | 0.0000 |
| 46 | 33.0348 | 1.1226 | 0.0000 |
| 47 | 33.7761 | 1.0930 | 0.0000 |
| 48 | 34.5174 | 1.0581 | 0.0000 |
| 49 | 35.2587 | 1.0252 | 0.0000 |
| 50 | 36.0000 | 1.0000 | 0.0000 |

BODY PANEL 4P

```
BODY PANEL ROUTINE

THERE ARE  15 TRANSVERSE VERTICAL PLANES THAT INTERSECT THE BODY
TO DEFINE PANEL LEADING AND TRAILING EDGES

X-INTERCEPTS

  13.76097
  14.75289
  15.69159
  16.61262
  17.53366
  18.45469
  19.37572
  20.29676
  21.21779
  22.13882
  23.05986
  25.00000
  27.00000
  29.50000
  32.41500
```

BODY PANEL CORNER POINT COORDINATES
1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

| PANEL NO | PARTS | X 1 | Y 1 | Z 1 | X 2 | Y 2 | Z 2 | X 3 | Y 3 | Z 3 | X 4 | Y 4 | Z 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 13.761 | 0.000 | 1.154 | 13.761 | 0.488 | 1.046 | 14.753 | 0.005 | 1.177 | 14.753 | 0.493 | 1.069 |
| 2 | 1 | 14.753 | 0.000 | 1.178 | 14.753 | 0.498 | 1.068 | 15.692 | 0.001 | 1.184 | 15.692 | 0.499 | 1.073 |
| 3 | 1 | 15.692 | 0.000 | 1.184 | 15.692 | 0.500 | 1.073 | 16.613 | 0.000 | 1.184 | 16.613 | 0.500 | 1.073 |
| 4 | 1 | 16.613 | 0.000 | 1.184 | 16.613 | 0.500 | 1.073 | 17.534 | 0.000 | 1.184 | 17.534 | 0.500 | 1.073 |
| 5 | 1 | 17.534 | 0.000 | 1.184 | 17.534 | 0.500 | 1.073 | 18.455 | 0.000 | 1.184 | 18.455 | 0.500 | 1.073 |
| 6 | 1 | 18.455 | 0.000 | 1.184 | 18.455 | 0.500 | 1.073 | 19.376 | 0.000 | 1.184 | 19.376 | 0.500 | 1.073 |
| 7 | 1 | 19.376 | 0.000 | 1.184 | 19.376 | 0.500 | 1.073 | 20.297 | 0.000 | 1.184 | 20.297 | 0.500 | 1.073 |
| 8 | 1 | 20.297 | 0.000 | 1.184 | 20.297 | 0.500 | 1.073 | 21.218 | 0.000 | 1.184 | 21.218 | 0.500 | 1.073 |
| 9 | 1 | 21.218 | 0.000 | 1.184 | 21.218 | 0.500 | 1.073 | 22.139 | 0.000 | 1.184 | 22.139 | 0.500 | 1.073 |
| 10 | 1 | 22.139 | 0.000 | 1.184 | 22.139 | 0.500 | 1.073 | 23.060 | 0.000 | 1.184 | 23.060 | 0.500 | 1.073 |
| 11 | 1 | 23.060 | 0.000 | 1.184 | 23.060 | 0.500 | 1.073 | 25.000 | 0.000 | 1.184 | 25.000 | 0.500 | 1.073 |
| 12 | 1 | 25.000 | 0.000 | 1.184 | 25.000 | 0.500 | 1.073 | 27.000 | 0.000 | 1.184 | 27.000 | 0.500 | 1.073 |
| 13 | 1 | 27.000 | 0.000 | 1.184 | 27.000 | 0.500 | 1.073 | 29.500 | 0.000 | 1.184 | 29.500 | 0.500 | 1.073 |
| 14 | 1 | 29.500 | 0.009 | 1.182 | 29.500 | 0.491 | 1.075 | 32.415 | 0.000 | 1.142 | 32.415 | 0.482 | 1.035 |
| 15 | 1 | 13.761 | 0.488 | 1.046 | 13.761 | 0.884 | 0.742 | 14.753 | 0.502 | 1.065 | 14.753 | 0.898 | 0.760 |
| 16 | 1 | 14.753 | 0.498 | 1.068 | 14.753 | 0.902 | 0.757 | 15.692 | 0.501 | 1.072 | 15.692 | 0.906 | 0.762 |
| 17 | 1 | 15.692 | 0.500 | 1.073 | 15.692 | 0.907 | 0.761 | 16.613 | 0.500 | 1.073 | 16.613 | 0.907 | 0.761 |
| 18 | 1 | 16.613 | 0.500 | 1.073 | 16.613 | 0.907 | 0.761 | 17.534 | 0.500 | 1.073 | 17.534 | 0.907 | 0.761 |
| 19 | 1 | 17.534 | 0.500 | 1.073 | 17.534 | 0.907 | 0.761 | 18.455 | 0.500 | 1.073 | 18.455 | 0.907 | 0.761 |
| 20 | 1 | 18.455 | 0.500 | 1.073 | 18.455 | 0.907 | 0.761 | 19.376 | 0.500 | 1.073 | 19.376 | 0.907 | 0.761 |
| 21 | 1 | 19.376 | 0.500 | 1.073 | 19.376 | 0.907 | 0.761 | 20.297 | 0.500 | 1.073 | 20.297 | 0.907 | 0.761 |
| 22 | 1 | 20.297 | 0.500 | 1.073 | 20.297 | 0.907 | 0.761 | 21.218 | 0.500 | 1.073 | 21.218 | 0.907 | 0.761 |
| 23 | 1 | 21.218 | 0.500 | 1.073 | 21.218 | 0.907 | 0.761 | 22.139 | 0.500 | 1.073 | 22.139 | 0.907 | 0.761 |
| 24 | 1 | 22.139 | 0.500 | 1.073 | 22.139 | 0.907 | 0.761 | 23.060 | 0.500 | 1.073 | 23.060 | 0.907 | 0.761 |
| 25 | 1 | 23.060 | 0.500 | 1.073 | 23.060 | 0.907 | 0.761 | 25.000 | 0.500 | 1.073 | 25.000 | 0.907 | 0.761 |
| 26 | 1 | 25.000 | 0.500 | 1.073 | 25.000 | 0.907 | 0.761 | 27.000 | 0.500 | 1.073 | 27.000 | 0.907 | 0.761 |
| 27 | 1 | 27.000 | 0.500 | 1.073 | 27.000 | 0.907 | 0.761 | 29.500 | 0.500 | 1.073 | 29.500 | 0.907 | 0.761 |
| 28 | 1 | 29.500 | 0.508 | 1.067 | 29.500 | 0.900 | 0.767 | 32.415 | 0.482 | 1.035 | 32.415 | 0.874 | 0.734 |
| 29 | 1 | 13.761 | 0.884 | 0.742 | 13.761 | 1.115 | 0.299 | 14.753 | 0.905 | 0.753 | 14.753 | 1.136 | 0.309 |
| 30 | 1 | 14.753 | 0.902 | 0.757 | 14.753 | 1.138 | 0.305 | 15.692 | 0.908 | 0.760 | 15.692 | 1.143 | 0.308 |
| 31 | 1 | 15.692 | 0.907 | 0.761 | 15.692 | 1.144 | 0.306 | 16.613 | 0.907 | 0.761 | 16.613 | 1.144 | 0.306 |
| 32 | 1 | 16.613 | 0.907 | 0.761 | 16.613 | 1.144 | 0.306 | 17.534 | 0.907 | 0.761 | 17.534 | 1.144 | 0.306 |
| 33 | 1 | 17.534 | 0.907 | 0.761 | 17.534 | 1.144 | 0.306 | 18.455 | 0.907 | 0.761 | 18.455 | 1.144 | 0.306 |
| 34 | 1 | 18.455 | 0.907 | 0.761 | 18.455 | 1.144 | 0.306 | 19.376 | 0.907 | 0.761 | 19.376 | 1.144 | 0.306 |
| 35 | 1 | 19.376 | 0.907 | 0.761 | 19.376 | 1.144 | 0.306 | 20.297 | 0.907 | 0.761 | 20.297 | 1.144 | 0.306 |
| 36 | 1 | 20.297 | 0.907 | 0.761 | 20.297 | 1.144 | 0.306 | 21.218 | 0.907 | 0.761 | 21.218 | 1.144 | 0.306 |
| 37 | 1 | 21.218 | 0.907 | 0.761 | 21.218 | 1.144 | 0.306 | 22.139 | 0.907 | 0.761 | 22.139 | 1.144 | 0.306 |
| 38 | 1 | 22.139 | 0.907 | 0.761 | 22.139 | 1.144 | 0.306 | 23.060 | 0.907 | 0.761 | 23.060 | 1.144 | 0.306 |
| 39 | 1 | 23.060 | 0.907 | 0.761 | 23.060 | 1.144 | 0.306 | 25.000 | 0.907 | 0.761 | 25.000 | 1.144 | 0.306 |
| 40 | 1 | 25.000 | 0.907 | 0.761 | 25.000 | 1.144 | 0.306 | 27.000 | 0.907 | 0.761 | 27.000 | 1.144 | 0.306 |
| 41 | 1 | 27.000 | 0.907 | 0.761 | 27.000 | 1.144 | 0.306 | 29.500 | 0.907 | 0.761 | 29.500 | 1.144 | 0.306 |
| 42 | 1 | 29.500 | 0.911 | 0.753 | 29.500 | 1.139 | 0.315 | 32.415 | 0.874 | 0.734 | 32.415 | 1.103 | 0.295 |
| 43 | 1 | 13.761 | 1.115 | 0.299 | 13.761 | 1.125 | -0.250 | 14.753 | 1.138 | 0.299 | 14.753 | 1.151 | -0.250 |
| 44 | 1 | 14.753 | 1.138 | 0.305 | 14.753 | 1.151 | -0.250 | 15.692 | 1.144 | 0.305 | 15.692 | 1.157 | -0.250 |
| 45 | 1 | 15.692 | 1.144 | 0.306 | 15.692 | 1.157 | -0.250 | 16.613 | 1.144 | 0.306 | 16.613 | 1.157 | -0.250 |
| 46 | 1 | 16.613 | 1.144 | 0.306 | 16.613 | 1.157 | -0.250 | 17.534 | 1.144 | 0.306 | 17.534 | 1.157 | -0.250 |
| 47 | 1 | 17.534 | 1.144 | 0.306 | 17.534 | 1.157 | -0.250 | 18.455 | 1.144 | 0.306 | 18.455 | 1.157 | -0.250 |
| 48 | 1 | 18.455 | 1.144 | 0.306 | 18.455 | 1.157 | -0.250 | 19.376 | 1.144 | 0.306 | 19.376 | 1.157 | -0.250 |
| 49 | 1 | 19.376 | 1.144 | 0.306 | 19.376 | 1.157 | -0.250 | 20.297 | 1.144 | 0.306 | 20.297 | 1.157 | -0.250 |
| 50 | 1 | 20.297 | 1.144 | 0.306 | 20.297 | 1.157 | -0.250 | 21.218 | 1.144 | 0.306 | 21.218 | 1.157 | -0.250 |
| 51 | 1 | 21.218 | 1.144 | 0.306 | 21.218 | 1.157 | -0.250 | 22.139 | 1.144 | 0.306 | 22.139 | 1.157 | -0.250 |
| 52 | 1 | 22.139 | 1.144 | 0.306 | 22.139 | 1.157 | -0.250 | 23.060 | 1.144 | 0.306 | 23.060 | 1.157 | -0.250 |
| 53 | 1 | 23.060 | 1.144 | 0.306 | 23.060 | 1.157 | -0.250 | 25.000 | 1.144 | 0.306 | 25.000 | 1.157 | -0.250 |
| 54 | 1 | 25.000 | 1.144 | 0.306 | 25.000 | 1.157 | -0.250 | 27.000 | 1.144 | 0.306 | 27.000 | 1.157 | -0.250 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 55 | 1 | 27.000 | 1.144 | 0.306 | 27.000 | 1.157 | -0.250 | 29.500 | 1.144 | 0.306 | 29.500 | 1.157 | -0.250 |
| 56 | 1 | 29.500 | 1.144 | 0.296 | 29.500 | 1.157 | -0.240 | 32.415 | 1.103 | 0.295 | 32.415 | 1.116 | -0.241 |
| 57 | 1 | 13.761 | 1.125 | -0.250 | 13.761 | 0.884 | -0.742 | 14.753 | 1.146 | -0.260 | 14.753 | 0.905 | -0.752 |
| 58 | 1 | 14.753 | 1.151 | -0.250 | 14.753 | 0.902 | -0.757 | 15.692 | 1.156 | -0.253 | 15.692 | 0.908 | -0.760 |
| 59 | 1 | 15.692 | 1.157 | -0.250 | 15.692 | 0.907 | -0.761 | 16.613 | 1.157 | -0.250 | 16.613 | 0.907 | -0.761 |
| 60 | 1 | 16.613 | 1.157 | -0.250 | 16.613 | 0.907 | -0.761 | 17.534 | 1.157 | -0.250 | 17.534 | 0.907 | -0.761 |
| 61 | 1 | 17.534 | 1.157 | -0.250 | 17.534 | 0.907 | -0.761 | 18.455 | 1.157 | -0.250 | 18.455 | 0.907 | -0.761 |
| 62 | 1 | 18.455 | 1.157 | -0.250 | 18.455 | 0.907 | -0.761 | 19.376 | 1.157 | -0.250 | 19.376 | 0.907 | -0.761 |
| 63 | 1 | 19.376 | 1.157 | -0.250 | 19.376 | 0.907 | -0.761 | 20.297 | 1.157 | -0.250 | 20.297 | 0.907 | -0.761 |
| 64 | 1 | 20.297 | 1.157 | -0.250 | 20.297 | 0.907 | -0.761 | 21.218 | 1.157 | -0.250 | 21.218 | 0.907 | -0.761 |
| 65 | 1 | 21.218 | 1.157 | -0.250 | 21.218 | 0.907 | -0.761 | 22.139 | 1.157 | -0.250 | 22.139 | 0.907 | -0.761 |
| 66 | 1 | 22.139 | 1.157 | -0.250 | 22.139 | 0.907 | -0.761 | 23.060 | 1.157 | -0.250 | 23.060 | 0.907 | -0.761 |
| 67 | 1 | 23.060 | 1.157 | -0.250 | 23.060 | 0.907 | -0.761 | 25.000 | 1.157 | -0.250 | 25.000 | 0.907 | -0.761 |
| 68 | 1 | 25.000 | 1.157 | -0.250 | 25.000 | 0.907 | -0.761 | 27.000 | 1.157 | -0.250 | 27.000 | 0.907 | -0.761 |
| 69 | 1 | 27.000 | 1.157 | -0.250 | 27.000 | 0.907 | -0.761 | 29.500 | 1.157 | -0.250 | 29.500 | 0.907 | -0.761 |
| 70 | 1 | 29.500 | 1.153 | -0.259 | 29.500 | 0.911 | -0.752 | 32.415 | 1.116 | -0.241 | 32.415 | 0.874 | -0.734 |
| 71 | 1 | 13.761 | 0.884 | -0.742 | 13.761 | 0.488 | -1.046 | 14.753 | 0.898 | -0.760 | 14.753 | 0.502 | -1.065 |
| 72 | 1 | 14.753 | 0.902 | -0.757 | 14.753 | 0.498 | -1.068 | 15.692 | 0.906 | -0.762 | 15.692 | 0.501 | -1.072 |
| 73 | 1 | 15.692 | 0.907 | -0.761 | 15.692 | 0.500 | -1.073 | 16.613 | 0.907 | -0.761 | 16.613 | 0.500 | -1.073 |
| 74 | 1 | 16.613 | 0.907 | -0.761 | 16.613 | 0.500 | -1.073 | 17.534 | 0.907 | -0.761 | 17.534 | 0.500 | -1.073 |
| 75 | 1 | 17.534 | 0.907 | -0.761 | 17.534 | 0.500 | -1.073 | 18.455 | 0.907 | -0.761 | 18.455 | 0.500 | -1.073 |
| 76 | 1 | 18.455 | 0.907 | -0.761 | 18.455 | 0.500 | -1.073 | 19.376 | 0.907 | -0.761 | 19.376 | 0.500 | -1.073 |
| 77 | 1 | 19.376 | 0.907 | -0.761 | 19.376 | 0.500 | -1.073 | 20.297 | 0.907 | -0.761 | 20.297 | 0.500 | -1.073 |
| 78 | 1 | 20.297 | 0.907 | -0.761 | 20.297 | 0.500 | -1.073 | 21.218 | 0.907 | -0.761 | 21.218 | 0.500 | -1.073 |
| 79 | 1 | 21.218 | 0.907 | -0.761 | 21.218 | 0.500 | -1.073 | 22.139 | 0.907 | -0.761 | 22.139 | 0.500 | -1.073 |
| 80 | 1 | 22.139 | 0.907 | -0.761 | 22.139 | 0.500 | -1.073 | 23.060 | 0.907 | -0.761 | 23.060 | 0.500 | -1.073 |
| 81 | 1 | 23.060 | 0.907 | -0.761 | 23.060 | 0.500 | -1.073 | 25.000 | 0.907 | -0.761 | 25.000 | 0.500 | -1.073 |
| 82 | 1 | 25.000 | 0.907 | -0.761 | 25.000 | 0.500 | -1.073 | 27.000 | 0.907 | -0.761 | 27.000 | 0.500 | -1.073 |
| 83 | 1 | 27.000 | 0.907 | -0.761 | 27.000 | 0.500 | -1.073 | 29.500 | 0.907 | -0.761 | 29.500 | 0.500 | -1.073 |
| 84 | 1 | 29.500 | 0.900 | -0.767 | 29.500 | 0.508 | -1.067 | 32.415 | 0.874 | -0.734 | 32.415 | 0.482 | -1.035 |
| 85 | 1 | 13.761 | 0.488 | -1.046 | 13.761 | 0.000 | -1.154 | 14.753 | 0.493 | -1.069 | 14.753 | 0.005 | -1.177 |
| 86 | 1 | 14.753 | 0.498 | -1.068 | 14.753 | 0.000 | -1.178 | 15.692 | 0.499 | -1.073 | 15.692 | 0.001 | -1.184 |
| 87 | 1 | 15.692 | 0.500 | -1.073 | 15.692 | 0.000 | -1.184 | 16.613 | 0.500 | -1.073 | 16.613 | 0.000 | -1.184 |
| 88 | 1 | 16.613 | 0.500 | -1.073 | 16.613 | 0.000 | -1.184 | 17.534 | 0.500 | -1.073 | 17.534 | 0.000 | -1.184 |
| 89 | 1 | 17.534 | 0.500 | -1.073 | 17.534 | 0.000 | -1.184 | 18.455 | 0.500 | -1.073 | 18.455 | 0.000 | -1.184 |
| 90 | 1 | 18.455 | 0.500 | -1.073 | 18.455 | 0.000 | -1.184 | 19.376 | 0.500 | -1.073 | 19.376 | 0.000 | -1.184 |
| 91 | 1 | 19.376 | 0.500 | -1.073 | 19.376 | 0.000 | -1.184 | 20.297 | 0.500 | -1.073 | 20.297 | 0.000 | -1.184 |
| 92 | 1 | 20.297 | 0.500 | -1.073 | 20.297 | 0.000 | -1.184 | 21.218 | 0.500 | -1.073 | 21.218 | 0.000 | -1.184 |
| 93 | 1 | 21.218 | 0.500 | -1.073 | 21.218 | 0.000 | -1.184 | 22.139 | 0.500 | -1.073 | 22.139 | 0.000 | -1.184 |
| 94 | 1 | 22.139 | 0.500 | -1.073 | 22.139 | 0.000 | -1.184 | 23.060 | 0.500 | -1.073 | 23.060 | 0.000 | -1.184 |
| 95 | 1 | 23.060 | 0.500 | -1.073 | 23.060 | 0.000 | -1.184 | 25.000 | 0.500 | -1.073 | 25.000 | 0.000 | -1.184 |
| 96 | 1 | 25.000 | 0.500 | -1.073 | 25.000 | 0.000 | -1.184 | 27.000 | 0.500 | -1.073 | 27.000 | 0.000 | -1.184 |
| 97 | 1 | 27.000 | 0.500 | -1.073 | 27.000 | 0.000 | -1.184 | 29.500 | 0.500 | -1.073 | 29.500 | 0.000 | -1.184 |
| 98 | 1 | 29.500 | 0.491 | -1.075 | 29.500 | 0.009 | -1.182 | 32.415 | 0.482 | -1.035 | 32.415 | 0.000 | -1.142 |

BODY PANEL CENTROID AND CONTROL POINT COORDINATES

| PANEL | X C | Y C | Z C | X CP | Y CP | Z CP | AREA | THETA-INCLIN | ALPHA-INCID |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 14.259 | 0.246 | 1.112 | 14.703 | 0.249 | 1.122 | 0.501 | -0.21817 | 0.02349 |
| 2 | 15.223 | 0.250 | 1.126 | 15.645 | 0.250 | 1.128 | 0.480 | -0.21817 | 0.00619 |
| 3 | 16.152 | 0.250 | 1.129 | 16.567 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 4 | 17.073 | 0.250 | 1.129 | 17.488 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 5 | 17.994 | 0.250 | 1.129 | 18.409 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 6 | 18.915 | 0.250 | 1.129 | 19.330 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 7 | 19.836 | 0.250 | 1.129 | 20.251 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 8 | 20.757 | 0.250 | 1.129 | 21.172 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 9 | 21.678 | 0.250 | 1.129 | 22.093 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 10 | 22.599 | 0.250 | 1.129 | 23.014 | 0.250 | 1.129 | 0.472 | -0.21817 | 0.00000 |
| 11 | 24.030 | 0.250 | 1.129 | 24.903 | 0.250 | 1.129 | 0.994 | -0.21817 | 0.00000 |
| 12 | 26.000 | 0.250 | 1.129 | 26.900 | 0.250 | 1.129 | 1.025 | -0.21817 | 0.00000 |
| 13 | 28.250 | 0.250 | 1.129 | 29.375 | 0.250 | 1.129 | 1.281 | -0.21817 | 0.00000 |
| 14 | 30.949 | 0.246 | 1.108 | 32.269 | 0.242 | 1.090 | 1.467 | -0.21817 | -0.01422 |
| 15 | 14.259 | 0.693 | 0.902 | 14.703 | 0.699 | 0.912 | 0.501 | -0.65450 | 0.02349 |
| 16 | 15.223 | 0.702 | 0.915 | 15.645 | 0.704 | 0.917 | 0.480 | -0.65450 | 0.00619 |
| 17 | 16.152 | 0.704 | 0.917 | 16.567 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 18 | 17.073 | 0.704 | 0.917 | 17.488 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 19 | 17.994 | 0.704 | 0.917 | 18.409 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 20 | 18.915 | 0.704 | 0.917 | 19.330 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 21 | 19.836 | 0.704 | 0.917 | 20.251 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 22 | 20.757 | 0.704 | 0.917 | 21.172 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 23 | 21.678 | 0.704 | 0.917 | 22.093 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 24 | 22.599 | 0.704 | 0.917 | 23.014 | 0.704 | 0.917 | 0.472 | -0.65450 | 0.00000 |
| 25 | 24.030 | 0.704 | 0.917 | 24.903 | 0.704 | 0.917 | 0.994 | -0.65450 | 0.00000 |
| 26 | 26.000 | 0.704 | 0.917 | 26.900 | 0.704 | 0.917 | 1.025 | -0.65450 | 0.00000 |
| 27 | 28.250 | 0.704 | 0.917 | 29.375 | 0.704 | 0.917 | 1.281 | -0.65450 | 0.00000 |
| 28 | 30.949 | 0.691 | 0.901 | 32.269 | 0.680 | 0.886 | 1.467 | -0.65450 | -0.01422 |
| 29 | 14.259 | 1.010 | 0.526 | 14.703 | 1.019 | 0.531 | 0.501 | -1.09083 | 0.02349 |
| 30 | 15.223 | 1.023 | 0.532 | 15.645 | 1.025 | 0.534 | 0.480 | -1.09083 | 0.00619 |
| 31 | 16.152 | 1.025 | 0.534 | 16.567 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 32 | 17.073 | 1.025 | 0.534 | 17.488 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 33 | 17.994 | 1.025 | 0.534 | 18.409 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 34 | 18.915 | 1.025 | 0.534 | 19.330 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 35 | 19.836 | 1.025 | 0.534 | 20.251 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 36 | 20.757 | 1.025 | 0.534 | 21.172 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 37 | 21.678 | 1.025 | 0.534 | 22.093 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 38 | 22.599 | 1.025 | 0.534 | 23.014 | 1.025 | 0.534 | 0.472 | -1.09083 | 0.00000 |
| 39 | 24.030 | 1.025 | 0.534 | 24.903 | 1.025 | 0.534 | 0.994 | -1.09083 | 0.00000 |
| 40 | 26.000 | 1.025 | 0.534 | 26.900 | 1.025 | 0.534 | 1.025 | -1.09083 | 0.00000 |
| 41 | 28.250 | 1.025 | 0.534 | 29.375 | 1.025 | 0.534 | 1.281 | -1.09083 | 0.00000 |
| 42 | 30.949 | 1.007 | 0.524 | 32.269 | 0.990 | 0.516 | 1.467 | -1.09083 | -0.01422 |
| 43 | 14.258 | 1.132 | 0.026 | 14.703 | 1.143 | 0.026 | 0.548 | -1.54979 | 0.02338 |
| 44 | 15.222 | 1.147 | 0.028 | 15.645 | 1.150 | 0.028 | 0.522 | -1.54686 | 0.00616 |
| 45 | 16.152 | 1.150 | 0.028 | 16.567 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 46 | 17.073 | 1.150 | 0.028 | 17.488 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 47 | 17.994 | 1.150 | 0.028 | 18.409 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 48 | 18.915 | 1.150 | 0.028 | 19.330 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 49 | 19.836 | 1.150 | 0.028 | 20.251 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 50 | 20.757 | 1.150 | 0.028 | 21.172 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 51 | 21.678 | 1.150 | 0.028 | 22.093 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 52 | 22.599 | 1.150 | 0.028 | 23.014 | 1.150 | 0.028 | 0.513 | -1.54628 | 0.00000 |
| 53 | 24.030 | 1.150 | 0.028 | 24.903 | 1.150 | 0.028 | 1.080 | -1.54628 | 0.00000 |
| 54 | 26.000 | 1.150 | 0.028 | 26.900 | 1.150 | 0.028 | 1.113 | -1.54627 | 0.00000 |
| 55 | 28.250 | 1.150 | 0.028 | 29.375 | 1.150 | 0.028 | 1.392 | -1.54627 | 0.00000 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 56 | 30.949 | 1.130 | 0.028 | 32.269 | 1.111 | 0.027 | 1.594 | -1.54627 | -0.01415 |
| 57 | 14.259 | 1.016 | -0.500 | 14.703 | 1.025 | -0.504 | 0.552 | -2.02624 | 0.02336 |
| 58 | 15.223 | 1.029 | -0.505 | 15.645 | 1.032 | -0.506 | 0.532 | -2.02624 | 0.00615 |
| 59 | 16.152 | 1.032 | -0.506 | 16.567 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 60 | 17.073 | 1.032 | -0.506 | 17.488 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 61 | 17.994 | 1.032 | -0.506 | 18.409 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 62 | 18.915 | 1.032 | -0.506 | 19.330 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 63 | 19.836 | 1.032 | -0.506 | 20.251 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 64 | 20.757 | 1.032 | -0.506 | 21.172 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 65 | 21.678 | 1.032 | -0.506 | 22.093 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 66 | 22.599 | 1.032 | -0.506 | 23.014 | 1.032 | -0.506 | 0.524 | -2.02624 | 0.00000 |
| 67 | 24.030 | 1.032 | -0.506 | 24.903 | 1.032 | -0.506 | 1.104 | -2.02624 | 0.00000 |
| 68 | 26.000 | 1.032 | -0.506 | 26.900 | 1.032 | -0.506 | 1.138 | -2.02624 | 0.00000 |
| 69 | 28.250 | 1.032 | -0.506 | 29.375 | 1.032 | -0.506 | 1.423 | -2.02624 | 0.00000 |
| 70 | 30.949 | 1.014 | -0.497 | 32.269 | 0.997 | -0.488 | 1.629 | -2.02624 | -0.01413 |
| 71 | 14.259 | 0.693 | -0.903 | 14.703 | 0.699 | -0.912 | 0.501 | -2.48709 | 0.02349 |
| 72 | 15.223 | 0.702 | -0.915 | 15.645 | 0.704 | -0.917 | 0.480 | -2.48709 | 0.00619 |
| 73 | 16.152 | 0.704 | -0.917 | 16.567 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 74 | 17.073 | 0.704 | -0.917 | 17.488 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 75 | 17.994 | 0.704 | -0.917 | 18.409 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 76 | 18.915 | 0.704 | -0.917 | 19.330 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 77 | 19.836 | 0.704 | -0.917 | 20.251 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 78 | 20.757 | 0.704 | -0.917 | 21.172 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 79 | 21.678 | 0.704 | -0.917 | 22.093 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 80 | 22.599 | 0.704 | -0.917 | 23.014 | 0.704 | -0.917 | 0.472 | -2.48709 | 0.00000 |
| 81 | 24.030 | 0.704 | -0.917 | 24.903 | 0.704 | -0.917 | 0.994 | -2.48709 | 0.00000 |
| 82 | 26.000 | 0.704 | -0.917 | 26.900 | 0.704 | -0.917 | 1.025 | -2.48709 | 0.00000 |
| 83 | 28.250 | 0.704 | -0.917 | 29.375 | 0.704 | -0.917 | 1.281 | -2.48709 | 0.00000 |
| 84 | 30.949 | 0.691 | -0.901 | 32.269 | 0.680 | -0.886 | 1.467 | -2.48709 | -0.01422 |
| 85 | 14.259 | 0.246 | -1.112 | 14.703 | 0.249 | -1.122 | 0.501 | -2.92343 | 0.02349 |
| 86 | 15.223 | 0.250 | -1.126 | 15.645 | 0.250 | -1.128 | 0.480 | -2.92343 | 0.00619 |
| 87 | 16.152 | 0.250 | -1.129 | 16.567 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 88 | 17.073 | 0.250 | -1.129 | 17.488 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 89 | 17.994 | 0.250 | -1.129 | 18.409 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 90 | 18.915 | 0.250 | -1.129 | 19.330 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 91 | 19.836 | 0.250 | -1.129 | 20.251 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 92 | 20.757 | 0.250 | -1.129 | 21.172 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 93 | 21.678 | 0.250 | -1.129 | 22.093 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 94 | 22.599 | 0.250 | -1.129 | 23.014 | 0.250 | -1.129 | 0.472 | -2.92343 | 0.00000 |
| 95 | 24.030 | 0.250 | -1.129 | 24.903 | 0.250 | -1.129 | 0.994 | -2.92343 | 0.00000 |
| 96 | 26.000 | 0.250 | -1.129 | 26.900 | 0.250 | -1.129 | 1.025 | -2.92343 | 0.00000 |
| 97 | 28.250 | 0.250 | -1.129 | 29.375 | 0.250 | -1.129 | 1.281 | -2.92343 | 0.00000 |
| 98 | 30.949 | 0.246 | -1.108 | 32.269 | 0.242 | -1.090 | 1.467 | -2.92343 | -0.01422 |

WING PANEL 8P

```
     WING PANEL ROUTINE

THERE ARE  10 VERTICAL PLANES THAT INTERSECT THE WING TO DEFINE PANEL SIDE EDGES

LEADING-EDGE
Y-INTERCEPT

   1.98460
   2.81190
   3.63920
   4.46650
   5.29380
   6.12110
   6.94840
   7.77570
   8.60300

          OUTBOARD CUTTING PLANE
       LEADING-EDGE
X-INTERCEPT      Y-INTERCEPT          SLOPE
 40.27000         10.77400             0.00000

ZERO-SLOPE INDICATES STREAMWISE CUTTING PLANE

UNDRFLOW AT 25422 IN MQ

UNDRFLOW AT 25422 IN MQ

UNDRFLOW AT 25422 IN MQ

AIRFOIL COORDINATES ARE USED DIRECTLY BY PROGRAM
```

WING PANEL CORNER POINT COORDINATES
1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

| PANEL NO | PARTS | X 1 | Y 1 | Z 1 | X 2 | Y 2 | Z 2 | X 3 | Y 3 | Z 3 | X 4 | Y 4 | Z 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 13.832 | 1.151 | -0.250 | 16.122 | 1.985 | -0.250 | 14.753 | 1.151 | -0.250 | 17.043 | 1.985 | -0.250 |
| 2 | 1 | 14.771 | 1.157 | -0.250 | 17.043 | 1.985 | -0.250 | 15.692 | 1.157 | -0.250 | 17.965 | 1.985 | -0.250 |
| 3 | 1 | 15.692 | 1.157 | -0.250 | 17.965 | 1.985 | -0.250 | 16.613 | 1.157 | -0.250 | 18.886 | 1.985 | -0.250 |
| 4 | 1 | 16.613 | 1.157 | -0.250 | 18.896 | 1.985 | -0.250 | 17.534 | 1.157 | -0.250 | 19.807 | 1.985 | -0.250 |
| 5 | 1 | 17.534 | 1.157 | -0.250 | 19.807 | 1.985 | -0.250 | 18.455 | 1.157 | -0.250 | 20.728 | 1.985 | -0.250 |
| 6 | 1 | 18.455 | 1.157 | -0.250 | 20.728 | 1.985 | -0.250 | 19.376 | 1.157 | -0.250 | 21.649 | 1.985 | -0.250 |
| 7 | 1 | 19.376 | 1.157 | -0.250 | 21.649 | 1.985 | -0.250 | 20.297 | 1.157 | -0.250 | 22.570 | 1.985 | -0.250 |
| 8 | 1 | 20.297 | 1.157 | -0.250 | 22.570 | 1.985 | -0.250 | 21.218 | 1.157 | -0.250 | 23.491 | 1.985 | -0.250 |
| 9 | 1 | 21.218 | 1.157 | -0.250 | 23.491 | 1.985 | -0.250 | 22.139 | 1.157 | -0.250 | 24.412 | 1.985 | -0.250 |
| 10 | 1 | 22.139 | 1.157 | -0.250 | 24.412 | 1.985 | -0.250 | 23.060 | 1.157 | -0.250 | 25.333 | 1.985 | -0.250 |
| 11 | 1 | 16.122 | 1.985 | -0.250 | 18.395 | 2.812 | -0.250 | 17.043 | 1.985 | -0.250 | 19.316 | 2.812 | -0.250 |
| 12 | 1 | 17.043 | 1.985 | -0.250 | 19.316 | 2.812 | -0.250 | 17.965 | 1.985 | -0.250 | 20.237 | 2.812 | -0.250 |
| 13 | 1 | 17.965 | 1.985 | -0.250 | 20.237 | 2.812 | -0.250 | 18.886 | 1.985 | -0.250 | 21.159 | 2.812 | -0.250 |
| 14 | 1 | 18.886 | 1.985 | -0.250 | 21.159 | 2.812 | -0.250 | 19.807 | 1.985 | -0.250 | 22.080 | 2.812 | -0.250 |
| 15 | 1 | 19.807 | 1.985 | -0.250 | 22.080 | 2.812 | -0.250 | 20.728 | 1.985 | -0.250 | 23.001 | 2.812 | -0.250 |
| 16 | 1 | 20.728 | 1.985 | -0.250 | 23.001 | 2.812 | -0.250 | 21.649 | 1.985 | -0.250 | 23.922 | 2.812 | -0.250 |
| 17 | 1 | 21.649 | 1.985 | -0.250 | 23.922 | 2.812 | -0.250 | 22.570 | 1.985 | -0.250 | 24.843 | 2.812 | -0.250 |
| 18 | 1 | 22.570 | 1.985 | -0.250 | 24.843 | 2.812 | -0.250 | 23.491 | 1.985 | -0.250 | 25.764 | 2.812 | -0.250 |
| 19 | 1 | 23.491 | 1.985 | -0.250 | 25.764 | 2.812 | -0.250 | 24.412 | 1.985 | -0.250 | 26.685 | 2.812 | -0.250 |
| 20 | 1 | 24.412 | 1.985 | -0.250 | 26.685 | 2.812 | -0.250 | 25.333 | 1.985 | -0.250 | 27.606 | 2.812 | -0.250 |
| 21 | 1 | 18.395 | 2.812 | -0.250 | 20.668 | 3.639 | -0.250 | 19.316 | 2.812 | -0.250 | 21.589 | 3.639 | -0.250 |
| 22 | 1 | 19.316 | 2.812 | -0.250 | 21.589 | 3.639 | -0.250 | 20.237 | 2.812 | -0.250 | 22.510 | 3.639 | -0.250 |
| 23 | 1 | 20.237 | 2.812 | -0.250 | 22.510 | 3.639 | -0.250 | 21.159 | 2.812 | -0.250 | 23.431 | 3.639 | -0.250 |
| 24 | 1 | 21.159 | 2.812 | -0.250 | 23.431 | 3.639 | -0.250 | 22.080 | 2.812 | -0.250 | 24.353 | 3.639 | -0.250 |
| 25 | 1 | 22.080 | 2.812 | -0.250 | 24.353 | 3.639 | -0.250 | 23.001 | 2.812 | -0.250 | 25.274 | 3.639 | -0.250 |
| 26 | 1 | 23.001 | 2.812 | -0.250 | 25.274 | 3.639 | -0.250 | 23.922 | 2.812 | -0.250 | 26.195 | 3.639 | -0.250 |
| 27 | 1 | 23.922 | 2.812 | -0.250 | 26.195 | 3.639 | -0.250 | 24.843 | 2.812 | -0.250 | 27.116 | 3.639 | -0.250 |
| 28 | 1 | 24.843 | 2.812 | -0.250 | 27.116 | 3.639 | -0.250 | 25.764 | 2.812 | -0.250 | 28.037 | 3.639 | -0.250 |
| 29 | 1 | 25.764 | 2.812 | -0.250 | 28.037 | 3.639 | -0.250 | 26.685 | 2.812 | -0.250 | 28.958 | 3.639 | -0.250 |
| 30 | 1 | 26.685 | 2.812 | -0.250 | 28.958 | 3.639 | -0.250 | 27.606 | 2.812 | -0.250 | 29.879 | 3.639 | -0.250 |
| 31 | 1 | 20.668 | 3.639 | -0.250 | 22.941 | 4.466 | -0.250 | 21.589 | 3.639 | -0.250 | 23.862 | 4.466 | -0.250 |
| 32 | 1 | 21.589 | 3.639 | -0.250 | 23.862 | 4.466 | -0.250 | 22.510 | 3.639 | -0.250 | 24.783 | 4.466 | -0.250 |
| 33 | 1 | 22.510 | 3.639 | -0.250 | 24.783 | 4.466 | -0.250 | 23.431 | 3.639 | -0.250 | 25.704 | 4.466 | -0.250 |
| 34 | 1 | 23.431 | 3.639 | -0.250 | 25.704 | 4.466 | -0.250 | 24.353 | 3.639 | -0.250 | 26.626 | 4.466 | -0.250 |
| 35 | 1 | 24.353 | 3.639 | -0.250 | 26.626 | 4.466 | -0.250 | 25.274 | 3.639 | -0.250 | 27.547 | 4.466 | -0.250 |
| 36 | 1 | 25.274 | 3.639 | -0.250 | 27.547 | 4.466 | -0.250 | 26.195 | 3.639 | -0.250 | 28.468 | 4.466 | -0.250 |
| 37 | 1 | 26.195 | 3.639 | -0.250 | 28.468 | 4.466 | -0.250 | 27.116 | 3.639 | -0.250 | 29.389 | 4.466 | -0.250 |
| 38 | 1 | 27.116 | 3.639 | -0.250 | 29.389 | 4.466 | -0.250 | 28.037 | 3.639 | -0.250 | 30.310 | 4.466 | -0.250 |
| 39 | 1 | 28.037 | 3.639 | -0.250 | 30.310 | 4.466 | -0.250 | 28.958 | 3.639 | -0.250 | 31.231 | 4.466 | -0.250 |
| 40 | 1 | 28.958 | 3.639 | -0.250 | 31.231 | 4.466 | -0.250 | 29.879 | 3.639 | -0.250 | 32.152 | 4.466 | -0.250 |
| 41 | 1 | 22.941 | 4.466 | -0.250 | 25.214 | 5.294 | -0.250 | 23.862 | 4.466 | -0.250 | 26.135 | 5.294 | -0.250 |
| 42 | 1 | 23.862 | 4.466 | -0.250 | 26.135 | 5.294 | -0.250 | 24.783 | 4.466 | -0.250 | 27.056 | 5.294 | -0.250 |
| 43 | 1 | 24.783 | 4.466 | -0.250 | 27.056 | 5.294 | -0.250 | 25.704 | 4.466 | -0.250 | 27.977 | 5.294 | -0.250 |
| 44 | 1 | 25.704 | 4.466 | -0.250 | 27.977 | 5.294 | -0.250 | 26.626 | 4.466 | -0.250 | 28.899 | 5.294 | -0.250 |
| 45 | 1 | 26.626 | 4.466 | -0.250 | 28.899 | 5.294 | -0.250 | 27.547 | 4.466 | -0.250 | 29.820 | 5.294 | -0.250 |
| 46 | 1 | 27.547 | 4.466 | -0.250 | 29.820 | 5.294 | -0.250 | 28.468 | 4.466 | -0.250 | 30.741 | 5.294 | -0.250 |
| 47 | 1 | 28.468 | 4.466 | -0.250 | 30.741 | 5.294 | -0.250 | 29.389 | 4.466 | -0.250 | 31.662 | 5.294 | -0.250 |
| 48 | 1 | 29.389 | 4.466 | -0.250 | 31.662 | 5.294 | -0.250 | 30.310 | 4.466 | -0.250 | 32.583 | 5.294 | -0.250 |
| 49 | 1 | 30.310 | 4.466 | -0.250 | 32.583 | 5.294 | -0.250 | 31.231 | 4.466 | -0.250 | 33.504 | 5.294 | -0.250 |
| 50 | 1 | 31.231 | 4.466 | -0.250 | 33.504 | 5.294 | -0.250 | 32.152 | 4.466 | -0.250 | 34.425 | 5.294 | -0.250 |
| 51 | 1 | 25.214 | 5.294 | -0.250 | 27.487 | 6.121 | -0.250 | 26.135 | 5.294 | -0.250 | 28.408 | 6.121 | -0.250 |
| 52 | 1 | 26.135 | 5.294 | -0.250 | 28.408 | 6.121 | -0.250 | 27.056 | 5.294 | -0.250 | 29.329 | 6.121 | -0.250 |
| 53 | 1 | 27.056 | 5.294 | -0.250 | 29.329 | 6.121 | -0.250 | 27.977 | 5.294 | -0.250 | 30.250 | 6.121 | -0.250 |
| 54 | 1 | 27.977 | 5.294 | -0.250 | 30.250 | 6.121 | -0.250 | 28.899 | 5.294 | -0.250 | 31.172 | 6.121 | -0.250 |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 55 | 1 | 28.899 | 5.294 | -0.250 | 31.172 | 6.121 | -0.250 | 29.820 | 5.294 | -0.250 | 32.093 | 6.121 | -0.250 |
| 56 | 1 | 29.820 | 5.294 | -0.250 | 32.093 | 6.121 | -0.250 | 30.741 | 5.294 | -0.250 | 33.014 | 6.121 | -0.250 |
| 57 | 1 | 30.741 | 5.294 | -0.250 | 33.014 | 6.121 | -0.250 | 31.662 | 5.294 | -0.250 | 33.935 | 6.121 | -0.250 |
| 58 | 1 | 31.662 | 5.294 | -0.250 | 33.935 | 6.121 | -0.250 | 32.583 | 5.294 | -0.250 | 34.856 | 6.121 | -0.250 |
| 59 | 1 | 32.583 | 5.294 | -0.250 | 34.856 | 6.121 | -0.250 | 33.504 | 5.294 | -0.250 | 35.777 | 6.121 | -0.250 |
| 60 | 1 | 33.504 | 5.294 | -0.250 | 35.777 | 6.121 | -0.250 | 34.425 | 5.294 | -0.250 | 36.699 | 6.121 | -0.250 |
| 61 | 1 | 27.487 | 6.121 | -0.250 | 29.760 | 6.948 | -0.250 | 28.408 | 6.121 | -0.250 | 30.681 | 6.948 | -0.250 |
| 62 | 1 | 28.408 | 6.121 | -0.250 | 30.681 | 6.948 | -0.250 | 29.329 | 6.121 | -0.250 | 31.602 | 6.948 | -0.250 |
| 63 | 1 | 29.329 | 6.121 | -0.250 | 31.602 | 6.948 | -0.250 | 30.250 | 6.121 | -0.250 | 32.523 | 6.948 | -0.250 |
| 64 | 1 | 30.250 | 6.121 | -0.250 | 32.523 | 6.948 | -0.250 | 31.172 | 6.121 | -0.250 | 33.445 | 6.948 | -0.250 |
| 65 | 1 | 31.172 | 6.121 | -0.250 | 33.445 | 6.948 | -0.250 | 32.093 | 6.121 | -0.250 | 34.366 | 6.948 | -0.250 |
| 66 | 1 | 32.093 | 6.121 | -0.250 | 34.366 | 6.948 | -0.250 | 33.014 | 6.121 | -0.250 | 35.287 | 6.948 | -0.250 |
| 67 | 1 | 33.014 | 6.121 | -0.250 | 35.287 | 6.948 | -0.250 | 33.935 | 6.121 | -0.250 | 36.208 | 6.948 | -0.250 |
| 68 | 1 | 33.935 | 6.121 | -0.250 | 36.208 | 6.948 | -0.250 | 34.856 | 6.121 | -0.250 | 37.129 | 6.948 | -0.250 |
| 69 | 1 | 34.856 | 6.121 | -0.250 | 37.129 | 6.948 | -0.250 | 35.777 | 6.121 | -0.250 | 38.051 | 6.948 | -0.250 |
| 70 | 1 | 35.777 | 6.121 | -0.250 | 38.051 | 6.948 | -0.250 | 36.699 | 6.121 | -0.250 | 38.972 | 6.948 | -0.250 |
| 71 | 1 | 29.760 | 6.948 | -0.250 | 32.033 | 7.776 | -0.250 | 30.681 | 6.948 | -0.250 | 32.954 | 7.776 | -0.250 |
| 72 | 1 | 30.681 | 6.948 | -0.250 | 32.954 | 7.776 | -0.250 | 31.602 | 6.948 | -0.250 | 33.875 | 7.776 | -0.250 |
| 73 | 1 | 31.602 | 6.948 | -0.250 | 33.875 | 7.776 | -0.250 | 32.523 | 6.948 | -0.250 | 34.796 | 7.776 | -0.250 |
| 74 | 1 | 32.523 | 6.948 | -0.250 | 34.796 | 7.776 | -0.250 | 33.445 | 6.948 | -0.250 | 35.718 | 7.776 | -0.250 |
| 75 | 1 | 33.445 | 6.948 | -0.250 | 35.718 | 7.776 | -0.250 | 34.366 | 6.948 | -0.250 | 36.639 | 7.776 | -0.250 |
| 76 | 1 | 34.366 | 6.948 | -0.250 | 36.639 | 7.776 | -0.250 | 35.287 | 6.948 | -0.250 | 37.560 | 7.776 | -0.250 |
| 77 | 1 | 35.287 | 6.948 | -0.250 | 37.560 | 7.776 | -0.250 | 36.208 | 6.948 | -0.250 | 38.481 | 7.776 | -0.250 |
| 78 | 1 | 36.208 | 6.948 | -0.250 | 38.481 | 7.776 | -0.250 | 37.129 | 6.948 | -0.250 | 39.402 | 7.776 | -0.250 |
| 79 | 1 | 37.129 | 6.948 | -0.250 | 39.402 | 7.776 | -0.250 | 38.051 | 6.948 | -0.250 | 40.324 | 7.776 | -0.250 |
| 80 | 1 | 38.051 | 6.948 | -0.250 | 40.324 | 7.776 | -0.250 | 38.972 | 6.948 | -0.250 | 41.245 | 7.776 | -0.250 |
| 81 | 1 | 32.033 | 7.776 | -0.250 | 34.305 | 8.603 | -0.250 | 32.954 | 7.776 | -0.250 | 35.227 | 8.603 | -0.250 |
| 82 | 1 | 32.954 | 7.776 | -0.250 | 35.227 | 8.603 | -0.250 | 33.875 | 7.776 | -0.250 | 36.148 | 8.603 | -0.250 |
| 83 | 1 | 33.875 | 7.776 | -0.250 | 36.148 | 8.603 | -0.250 | 34.796 | 7.776 | -0.250 | 37.069 | 8.603 | -0.250 |
| 84 | 1 | 34.796 | 7.776 | -0.250 | 37.069 | 8.603 | -0.250 | 35.718 | 7.776 | -0.250 | 37.990 | 8.603 | -0.250 |
| 85 | 1 | 35.718 | 7.776 | -0.250 | 37.990 | 8.603 | -0.250 | 36.639 | 7.776 | -0.250 | 38.912 | 8.603 | -0.250 |
| 86 | 1 | 36.639 | 7.776 | -0.250 | 38.912 | 8.603 | -0.250 | 37.560 | 7.776 | -0.250 | 39.833 | 8.603 | -0.250 |
| 87 | 1 | 37.560 | 7.776 | -0.250 | 39.833 | 8.603 | -0.250 | 38.481 | 7.776 | -0.250 | 40.754 | 8.603 | -0.250 |
| 88 | 1 | 38.481 | 7.776 | -0.250 | 40.754 | 8.603 | -0.250 | 39.402 | 7.776 | -0.250 | 41.675 | 8.603 | -0.250 |
| 89 | 1 | 39.402 | 7.776 | -0.250 | 41.675 | 8.603 | -0.250 | 40.324 | 7.776 | -0.250 | 42.597 | 8.603 | -0.250 |
| 90 | 1 | 40.324 | 7.776 | -0.250 | 42.597 | 8.603 | -0.250 | 41.245 | 7.776 | -0.250 | 43.518 | 8.603 | -0.250 |
| 91 | 1 | 34.305 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 35.227 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 92 | 1 | 35.227 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 36.148 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 93 | 1 | 36.148 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 37.069 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 94 | 1 | 37.069 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 37.990 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 95 | 1 | 37.990 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 38.912 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 96 | 1 | 38.912 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 39.833 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 97 | 1 | 39.833 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 40.754 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 98 | 1 | 40.754 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 41.675 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 99 | 1 | 41.675 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 42.597 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |
| 100 | 1 | 42.597 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 | 43.518 | 8.603 | -0.250 | 40.270 | 10.774 | -0.250 |

WING PANEL CENTROID AND CONTROL POINT COORDINATES

| PANEL | X C | Y C | Z C | X CP | Y CP | Z CP | AREA | Z THICK | ALPHA-THICK | Z CAMBER | ALPHA-CAMBER |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 15.418 | 1.561 | -0.250 | 15.834 | 1.561 | -0.250 | 0.780 | -0.13106 | 0.06738 | -0.17193 | 0.04620 |
| 2 | 16.363 | 1.569 | -0.250 | 16.778 | 1.569 | -0.250 | 0.765 | -0.09283 | 0.03510 | -0.14713 | 0.01662 |
| 3 | 17.289 | 1.571 | -0.250 | 17.703 | 1.571 | -0.250 | 0.762 | -0.06967 | 0.01634 | -0.13461 | 0.01069 |
| 4 | 18.210 | 1.571 | -0.250 | 18.624 | 1.571 | -0.250 | 0.762 | -0.05866 | 0.00011 | -0.12665 | 0.00483 |
| 5 | 19.131 | 1.571 | -0.250 | 19.545 | 1.571 | -0.250 | 0.762 | -0.06419 | -0.01523 | -0.12404 | -0.00126 |
| 6 | 20.052 | 1.571 | -0.250 | 20.466 | 1.571 | -0.250 | 0.762 | -0.08510 | -0.02923 | -0.12792 | -0.00781 |
| 7 | 20.973 | 1.571 | -0.250 | 21.387 | 1.571 | -0.250 | 0.762 | -0.11931 | -0.03690 | -0.14046 | -0.01879 |
| 8 | 21.894 | 1.571 | -0.250 | 22.308 | 1.571 | -0.250 | 0.762 | -0.15869 | -0.04413 | -0.16352 | -0.02996 |
| 9 | 22.815 | 1.571 | -0.250 | 23.229 | 1.571 | -0.250 | 0.762 | -0.20009 | -0.05137 | -0.19806 | -0.04320 |
| 10 | 23.736 | 1.571 | -0.250 | 24.150 | 1.571 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 11 | 17.719 | 2.398 | -0.250 | 18.134 | 2.398 | -0.250 | 0.762 | -0.13105 | 0.06738 | -0.17192 | 0.04620 |
| 12 | 18.640 | 2.398 | -0.250 | 19.055 | 2.398 | -0.250 | 0.762 | -0.09283 | 0.03510 | -0.14713 | 0.01662 |
| 13 | 19.562 | 2.398 | -0.250 | 19.976 | 2.398 | -0.250 | 0.762 | -0.06966 | 0.01634 | -0.13461 | 0.01069 |
| 14 | 20.483 | 2.398 | -0.250 | 20.897 | 2.398 | -0.250 | 0.762 | -0.05866 | 0.00011 | -0.12665 | 0.00483 |
| 15 | 21.404 | 2.398 | -0.250 | 21.818 | 2.398 | -0.250 | 0.762 | -0.06418 | -0.01523 | -0.12403 | -0.00126 |
| 16 | 22.325 | 2.398 | -0.250 | 22.739 | 2.398 | -0.250 | 0.762 | -0.08509 | -0.02923 | -0.12792 | -0.00781 |
| 17 | 23.246 | 2.398 | -0.250 | 23.660 | 2.398 | -0.250 | 0.762 | -0.11931 | -0.03690 | -0.14046 | -0.01879 |
| 18 | 24.167 | 2.398 | -0.250 | 24.581 | 2.398 | -0.250 | 0.762 | -0.15868 | -0.04413 | -0.16351 | -0.02996 |
| 19 | 25.088 | 2.398 | -0.250 | 25.502 | 2.398 | -0.250 | 0.762 | -0.20009 | -0.05137 | -0.19806 | -0.04320 |
| 20 | 26.009 | 2.398 | -0.250 | 26.423 | 2.398 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 21 | 19.992 | 3.226 | -0.250 | 20.407 | 3.226 | -0.250 | 0.762 | -0.13105 | 0.06738 | -0.17192 | 0.04620 |
| 22 | 20.913 | 3.226 | -0.250 | 21.328 | 3.226 | -0.250 | 0.762 | -0.09282 | 0.03510 | -0.14713 | 0.01662 |
| 23 | 21.834 | 3.226 | -0.250 | 22.249 | 3.226 | -0.250 | 0.762 | -0.06966 | 0.01634 | -0.13460 | 0.01069 |
| 24 | 22.756 | 3.226 | -0.250 | 23.170 | 3.226 | -0.250 | 0.762 | -0.05865 | 0.00011 | -0.12664 | 0.00483 |
| 25 | 23.677 | 3.226 | -0.250 | 24.091 | 3.226 | -0.250 | 0.762 | -0.06418 | -0.01523 | -0.12403 | -0.00126 |
| 26 | 24.598 | 3.226 | -0.250 | 25.012 | 3.226 | -0.250 | 0.762 | -0.08509 | -0.02923 | -0.12792 | -0.00781 |
| 27 | 25.519 | 3.226 | -0.250 | 25.933 | 3.226 | -0.250 | 0.762 | -0.11931 | -0.03690 | -0.14046 | -0.01879 |
| 28 | 26.440 | 3.226 | -0.250 | 26.854 | 3.226 | -0.250 | 0.762 | -0.15868 | -0.04413 | -0.16351 | -0.02996 |
| 29 | 27.361 | 3.226 | -0.250 | 27.776 | 3.226 | -0.250 | 0.762 | -0.20009 | -0.05137 | -0.19806 | -0.04320 |
| 30 | 28.282 | 3.226 | -0.250 | 28.697 | 3.226 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 31 | 22.265 | 4.053 | -0.250 | 22.680 | 4.053 | -0.250 | 0.762 | -0.13105 | 0.06738 | -0.17192 | 0.04620 |
| 32 | 23.186 | 4.053 | -0.250 | 23.601 | 4.053 | -0.250 | 0.762 | -0.09282 | 0.03510 | -0.14712 | 0.01662 |
| 33 | 24.107 | 4.053 | -0.250 | 24.522 | 4.053 | -0.250 | 0.762 | -0.06965 | 0.01634 | -0.13460 | 0.01069 |
| 34 | 25.029 | 4.053 | -0.250 | 25.443 | 4.053 | -0.250 | 0.762 | -0.05865 | 0.00011 | -0.12664 | 0.00483 |
| 35 | 25.950 | 4.053 | -0.250 | 26.364 | 4.053 | -0.250 | 0.762 | -0.06417 | -0.01523 | -0.12403 | -0.00126 |
| 36 | 26.871 | 4.053 | -0.250 | 27.285 | 4.053 | -0.250 | 0.762 | -0.08508 | -0.02923 | -0.12791 | -0.00781 |
| 37 | 27.792 | 4.053 | -0.250 | 28.206 | 4.053 | -0.250 | 0.762 | -0.11930 | -0.03690 | -0.14045 | -0.01879 |
| 38 | 28.713 | 4.053 | -0.250 | 29.128 | 4.053 | -0.250 | 0.762 | -0.15868 | -0.04413 | -0.16351 | -0.02996 |
| 39 | 29.634 | 4.053 | -0.250 | 30.049 | 4.053 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 40 | 30.555 | 4.053 | -0.250 | 30.970 | 4.053 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 41 | 24.538 | 4.880 | -0.250 | 24.953 | 4.880 | -0.250 | 0.762 | -0.13104 | 0.06738 | -0.17192 | 0.04620 |
| 42 | 25.459 | 4.880 | -0.250 | 25.874 | 4.880 | -0.250 | 0.762 | -0.09282 | 0.03510 | -0.14712 | 0.01662 |
| 43 | 26.380 | 4.880 | -0.250 | 26.795 | 4.880 | -0.250 | 0.762 | -0.06965 | 0.01634 | -0.13460 | 0.01069 |
| 44 | 27.302 | 4.880 | -0.250 | 27.716 | 4.880 | -0.250 | 0.762 | -0.05864 | 0.00011 | -0.12664 | 0.00483 |
| 45 | 28.223 | 4.880 | -0.250 | 28.637 | 4.880 | -0.250 | 0.762 | -0.06417 | -0.01523 | -0.12402 | -0.00126 |
| 46 | 29.144 | 4.880 | -0.250 | 29.558 | 4.880 | -0.250 | 0.762 | -0.08508 | -0.02923 | -0.12791 | -0.00781 |
| 47 | 30.065 | 4.880 | -0.250 | 30.479 | 4.880 | -0.250 | 0.762 | -0.11930 | -0.03690 | -0.14045 | -0.01879 |
| 48 | 30.986 | 4.880 | -0.250 | 31.401 | 4.880 | -0.250 | 0.762 | -0.15868 | -0.04413 | -0.16351 | -0.02996 |
| 49 | 31.907 | 4.880 | -0.250 | 32.322 | 4.880 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 50 | 32.828 | 4.880 | -0.250 | 33.243 | 4.880 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 51 | 26.811 | 5.707 | -0.250 | 27.226 | 5.707 | -0.250 | 0.762 | -0.13104 | 0.06738 | -0.17192 | 0.04620 |
| 52 | 27.732 | 5.707 | -0.250 | 28.147 | 5.707 | -0.250 | 0.762 | -0.09281 | 0.03510 | -0.14712 | 0.01662 |
| 53 | 28.653 | 5.707 | -0.250 | 29.068 | 5.707 | -0.250 | 0.762 | -0.06964 | 0.01634 | -0.13459 | 0.01069 |
| 54 | 29.574 | 5.707 | -0.250 | 29.989 | 5.707 | -0.250 | 0.762 | -0.05864 | 0.00011 | -0.12663 | 0.00483 |
| 55 | 30.496 | 5.707 | -0.250 | 30.910 | 5.707 | -0.250 | 0.762 | -0.06416 | -0.01523 | -0.12402 | -0.00126 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 56 | 31.417 | 5.707 | -0.250 | 31.831 | 5.707 | -0.250 | 0.762 | -0.08507 | -0.02923 | -0.12791 | -0.00781 |
| 57 | 32.338 | 5.707 | -0.250 | 32.752 | 5.707 | -0.250 | 0.762 | -0.11930 | -0.03690 | -0.14045 | -0.01879 |
| 58 | 33.259 | 5.707 | -0.250 | 33.674 | 5.707 | -0.250 | 0.762 | -0.15868 | -0.04413 | -0.16351 | -0.02996 |
| 59 | 34.180 | 5.707 | -0.250 | 34.595 | 5.707 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 60 | 35.101 | 5.707 | -0.250 | 35.516 | 5.707 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 61 | 29.084 | 6.535 | -0.250 | 29.498 | 6.535 | -0.250 | 0.762 | -0.13104 | 0.06738 | -0.17191 | 0.04620 |
| 62 | 30.005 | 6.535 | -0.250 | 30.420 | 6.535 | -0.250 | 0.762 | -0.09281 | 0.03510 | -0.14712 | 0.01662 |
| 63 | 30.926 | 6.535 | -0.250 | 31.341 | 6.535 | -0.250 | 0.762 | -0.06964 | 0.01634 | -0.13459 | 0.01069 |
| 64 | 31.847 | 6.535 | -0.250 | 32.262 | 6.535 | -0.250 | 0.762 | -0.05863 | 0.00011 | -0.12663 | 0.00483 |
| 65 | 32.769 | 6.535 | -0.250 | 33.183 | 6.535 | -0.250 | 0.762 | -0.06416 | -0.01523 | -0.12402 | -0.00126 |
| 66 | 33.690 | 6.535 | -0.250 | 34.104 | 6.535 | -0.250 | 0.762 | -0.08507 | -0.02923 | -0.12790 | -0.00781 |
| 67 | 34.611 | 6.535 | -0.250 | 35.026 | 6.535 | -0.250 | 0.762 | -0.11929 | -0.03690 | -0.14045 | -0.01879 |
| 68 | 35.532 | 6.535 | -0.250 | 35.947 | 6.535 | -0.250 | 0.762 | -0.15867 | -0.04413 | -0.16350 | -0.02996 |
| 69 | 36.453 | 6.535 | -0.250 | 36.868 | 6.535 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 70 | 37.375 | 6.535 | -0.250 | 37.789 | 6.535 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24687 | -0.06070 |
| 71 | 31.357 | 7.362 | -0.250 | 31.771 | 7.362 | -0.250 | 0.762 | -0.13103 | 0.06738 | -0.17191 | 0.04620 |
| 72 | 32.278 | 7.362 | -0.250 | 32.693 | 7.362 | -0.250 | 0.762 | -0.09280 | 0.03510 | -0.14711 | 0.01662 |
| 73 | 33.199 | 7.362 | -0.250 | 33.614 | 7.362 | -0.250 | 0.762 | -0.06964 | 0.01634 | -0.13459 | 0.01069 |
| 74 | 34.120 | 7.362 | -0.250 | 34.535 | 7.362 | -0.250 | 0.762 | -0.05863 | 0.00011 | -0.12663 | 0.00483 |
| 75 | 35.042 | 7.362 | -0.250 | 35.456 | 7.362 | -0.250 | 0.762 | -0.06415 | -0.01523 | -0.12401 | -0.00126 |
| 76 | 35.963 | 7.362 | -0.250 | 36.377 | 7.362 | -0.250 | 0.762 | -0.08507 | -0.02923 | -0.12790 | -0.00781 |
| 77 | 36.884 | 7.362 | -0.250 | 37.299 | 7.362 | -0.250 | 0.762 | -0.11929 | -0.03690 | -0.14044 | -0.01879 |
| 78 | 37.805 | 7.362 | -0.250 | 38.220 | 7.362 | -0.250 | 0.762 | -0.15867 | -0.04413 | -0.16350 | -0.02996 |
| 79 | 38.726 | 7.362 | -0.250 | 39.141 | 7.362 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 80 | 39.648 | 7.362 | -0.250 | 40.062 | 7.362 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24686 | -0.06070 |
| 81 | 33.630 | 8.189 | -0.250 | 34.044 | 8.189 | -0.250 | 0.762 | -0.13103 | 0.06738 | -0.17191 | 0.04620 |
| 82 | 34.551 | 8.189 | -0.250 | 34.965 | 8.189 | -0.250 | 0.762 | -0.09280 | 0.03510 | -0.14711 | 0.01662 |
| 83 | 35.472 | 8.189 | -0.250 | 35.887 | 8.189 | -0.250 | 0.762 | -0.06963 | 0.01634 | -0.13458 | 0.01069 |
| 84 | 36.393 | 8.189 | -0.250 | 36.808 | 8.189 | -0.250 | 0.762 | -0.05862 | 0.00011 | -0.12662 | 0.00483 |
| 85 | 37.315 | 8.189 | -0.250 | 37.729 | 8.189 | -0.250 | 0.762 | -0.06415 | -0.01523 | -0.12401 | -0.00126 |
| 86 | 38.236 | 8.189 | -0.250 | 38.650 | 8.189 | -0.250 | 0.762 | -0.08506 | -0.02923 | -0.12790 | -0.00781 |
| 87 | 39.157 | 8.189 | -0.250 | 39.572 | 8.189 | -0.250 | 0.762 | -0.11929 | -0.03690 | -0.14044 | -0.01879 |
| 88 | 40.078 | 8.189 | -0.250 | 40.493 | 8.189 | -0.250 | 0.762 | -0.15867 | -0.04413 | -0.16350 | -0.02996 |
| 89 | 41.000 | 8.189 | -0.250 | 41.414 | 8.189 | -0.250 | 0.762 | -0.20008 | -0.05137 | -0.19805 | -0.04320 |
| 90 | 41.921 | 8.189 | -0.250 | 42.335 | 8.189 | -0.250 | 0.762 | -0.25001 | -0.05876 | -0.24686 | -0.06070 |
| 91 | 36.601 | 9.327 | -0.250 | 36.877 | 9.327 | -0.250 | 1.000 | -0.17069 | 0.06738 | -0.19794 | 0.04620 |
| 92 | 37.215 | 9.327 | -0.250 | 37.491 | 9.327 | -0.250 | 1.000 | -0.14520 | 0.03510 | -0.18141 | 0.01662 |
| 93 | 37.829 | 9.327 | -0.250 | 38.105 | 9.327 | -0.250 | 1.000 | -0.12976 | 0.01634 | -0.17306 | 0.01069 |
| 94 | 38.443 | 9.327 | -0.250 | 38.720 | 9.327 | -0.250 | 1.000 | -0.12242 | 0.00011 | -0.16775 | 0.00483 |
| 95 | 39.057 | 9.327 | -0.250 | 39.334 | 9.327 | -0.250 | 1.000 | -0.12610 | -0.01523 | -0.16601 | -0.00126 |
| 96 | 39.672 | 9.327 | -0.250 | 39.948 | 9.327 | -0.250 | 1.000 | -0.14004 | -0.02923 | -0.16860 | -0.00781 |
| 97 | 40.286 | 9.327 | -0.250 | 40.562 | 9.327 | -0.250 | 1.000 | -0.16286 | -0.03690 | -0.17696 | -0.01879 |
| 98 | 40.900 | 9.327 | -0.250 | 41.176 | 9.327 | -0.250 | 1.000 | -0.18911 | -0.04413 | -0.19233 | -0.02996 |
| 99 | 41.514 | 9.327 | -0.250 | 41.790 | 9.327 | -0.250 | 1.000 | -0.21672 | -0.05137 | -0.21537 | -0.04320 |
| 100 | 42.128 | 9.327 | -0.250 | 42.405 | 9.327 | -0.250 | 1.000 | -0.25001 | -0.05876 | -0.24791 | -0.06070 |

PANEND 14P

AERODYNAMIC 1A TIME 01.599 052067

SAVE TAPE

```
DESCRIPTION OF CASE REQUESTED

SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH SIDES OF X-Z PLANE(SYM = 1.)

CASE = 2.          CALCULATE CL, GIVEN SHAPE

CPCALC = 0.        LINEAR CP

POLAR = 0.         POLARS NOT REQUESTED

THICK = 1.         WING THICKNESS PRESSURES TO BE ADDED

VOUT = 1.          VELOCITY COMPONENTS TO BE PRINTED

MACH NUMBER =    1.8000

POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED
 X-COORDINATE =    0.0000
 Z-COORDINATE =    0.0000

REFERENCE CHORD LENGTH =    1.0000

WING REFERENCE AREA =   89.3750

WING SEMI-SPAN =     1.0000


GIVEN BODY RADII


GIVEN BODY CAMBER

HEIGHT OF WING PLANE ABOVE BODY AXIS =   -0.2500

INCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AXIS =    0.0000  DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS =    6.0000  DEG.


CONSTANT WING CAMBER


GIVEN WING THICKNESS
```

BODY GEOMETRY

| X-STATION | CAMBER | RADIUS | FIRST DERIVATIVE | SECOND DERIVATIVE |
|---|---|---|---|---|
| 0.0000 | 0.0000 | 0.0000 | 0.1800 | -0.0199 |
| 0.7243 | 0.0000 | 0.1304 | 0.1872 | 0.0398 |
| 1.4485 | 0.0000 | 0.2607 | 0.1512 | -0.1393 |
| 2.1728 | 0.0000 | 0.3491 | 0.1142 | 0.0372 |
| 2.8970 | 0.0000 | 0.4343 | 0.1110 | -0.0461 |
| 3.6213 | 0.0000 | 0.5077 | 0.0987 | 0.0121 |
| 4.3456 | 0.0000 | 0.5791 | 0.0941 | -0.0247 |
| 5.0698 | 0.0000 | 0.6433 | 0.0869 | 0.0048 |
| 5.7941 | 0.0000 | 0.7057 | 0.0826 | -0.0168 |
| 6.5184 | 0.0000 | 0.7627 | 0.0769 | 0.0013 |
| 7.2426 | 0.0000 | 0.8175 | 0.0728 | -0.0126 |
| 7.9669 | 0.0000 | 0.8679 | 0.0678 | -0.0015 |
| 8.6911 | 0.0000 | 0.9159 | 0.0638 | -0.0094 |
| 9.4154 | 0.0000 | 0.9602 | 0.0591 | -0.0037 |
| 10.1397 | 0.0000 | 1.0017 | 0.0550 | -0.0075 |
| 10.8639 | 0.0000 | 1.0396 | 0.0498 | -0.0067 |
| 11.5882 | 0.0000 | 1.0740 | 0.0454 | -0.0056 |
| 12.3124 | 0.0000 | 1.1049 | 0.0392 | -0.0113 |
| 13.0367 | 0.0000 | 1.1311 | 0.0342 | -0.0027 |
| 13.7610 | 0.0000 | 1.1542 | 0.0281 | -0.0142 |
| 14.5023 | 0.0000 | 1.1720 | 0.0213 | -0.0039 |
| 15.2436 | 0.0000 | 1.1840 | 0.0073 | -0.0340 |
| 15.9849 | 0.0000 | 1.1840 | -0.0019 | 0.0091 |
| 16.7262 | 0.0000 | 1.1840 | 0.0005 | -0.0024 |
| 17.4675 | 0.0000 | 1.1840 | -0.0001 | 0.0007 |
| 18.2088 | 0.0000 | 1.1840 | 0.0000 | -0.0002 |
| 18.9501 | 0.0000 | 1.1840 | -0.0000 | 0.0000 |
| 19.6914 | 0.0000 | 1.1840 | 0.0000 | -0.0000 |
| 20.4327 | 0.0000 | 1.1840 | -0.0000 | 0.0000 |
| 21.1740 | 0.0000 | 1.1840 | 0.0000 | -0.0000 |
| 21.9153 | 0.0000 | 1.1840 | -0.0000 | 0.0000 |
| 22.6566 | 0.0000 | 1.1840 | 0.0000 | -0.0000 |
| 23.3979 | 0.0000 | 1.1840 | -0.0000 | -0.0000 |
| 24.1392 | 0.0000 | 1.1840 | 0.0000 | 0.0000 |
| 24.8805 | 0.0000 | 1.1840 | -0.0000 | -0.0000 |
| 25.6218 | 0.0000 | 1.1840 | 0.0000 | 0.0000 |
| 26.3631 | 0.0000 | 1.1840 | -0.0000 | -0.0000 |
| 27.1044 | 0.0000 | 1.1840 | 0.0000 | 0.0000 |
| 27.8457 | 0.0000 | 1.1840 | -0.0000 | -0.0001 |
| 28.5870 | 0.0000 | 1.1840 | -0.0001 | 0.0004 |
| 29.3283 | 0.0000 | 1.1840 | -0.0003 | -0.0016 |
| 30.0696 | 0.0000 | 1.1825 | -0.0050 | -0.0109 |
| 30.8109 | 0.0000 | 1.1757 | -0.0133 | -0.0115 |
| 31.5522 | 0.0000 | 1.1630 | -0.0208 | -0.0090 |
| 32.2935 | 0.0000 | 1.1452 | -0.0269 | -0.0073 |
| 33.0348 | 0.0000 | 1.1226 | -0.0350 | -0.0146 |
| 33.7761 | 0.0000 | 1.0930 | -0.0446 | -0.0114 |
| 34.5174 | 0.0000 | 1.0581 | -0.0475 | 0.0038 |
| 35.2587 | 0.0000 | 1.0252 | -0.0398 | 0.0170 |

VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

| THETA(DEG.) X | 0.0000 | 22.5000 | 45.0000 | 67.5000 | 90.0000 | 112.5000 | 135.0000 | 157.5000 | 180.0000 |
|---|---|---|---|---|---|---|---|---|---|
| 0.0000 | -0.03166 | -0.03402 | -0.04073 | -0.05078 | -0.06263 | -0.07449 | -0.08454 | -0.09125 | -0.09361 |
| 0.7243 | -0.03446 | -0.03682 | -0.04353 | -0.05358 | -0.06543 | -0.07729 | -0.08734 | -0.09405 | -0.09641 |
| 1.4485 | -0.01686 | -0.01926 | -0.02608 | -0.03630 | -0.04836 | -0.06041 | -0.07063 | -0.07746 | -0.07986 |
| 2.1728 | -0.00369 | -0.00550 | -0.01066 | -0.01838 | -0.02749 | -0.03659 | -0.04431 | -0.04947 | -0.05128 |
| 2.8970 | -0.00659 | -0.00832 | -0.01323 | -0.02058 | -0.02925 | -0.03792 | -0.04527 | -0.05018 | -0.05190 |
| 3.6213 | -0.00198 | -0.00352 | -0.00793 | -0.01453 | -0.02230 | -0.03008 | -0.03668 | -0.04109 | -0.04263 |
| 4.3456 | -0.00193 | -0.00341 | -0.00764 | -0.01396 | -0.02142 | -0.02888 | -0.03520 | -0.03943 | -0.04091 |
| 5.0698 | 0.00027 | -0.00112 | -0.00505 | -0.01095 | -0.01790 | -0.02485 | -0.03075 | -0.03468 | -0.03607 |
| 5.7941 | 0.00087 | -0.00046 | -0.00423 | -0.00987 | -0.01653 | -0.02320 | -0.02884 | -0.03261 | -0.03394 |
| 6.5184 | 0.00239 | 0.00114 | -0.00241 | -0.00773 | -0.01400 | -0.02027 | -0.02559 | -0.02914 | -0.03039 |
| 7.2426 | 0.00305 | 0.00186 | -0.00153 | -0.00661 | -0.01260 | -0.01859 | -0.02366 | -0.02706 | -0.02825 |
| 7.9669 | 0.00434 | 0.00322 | 0.00002 | -0.00477 | -0.01042 | -0.01606 | -0.02085 | -0.02405 | -0.02517 |
| 8.6911 | 0.00494 | 0.00387 | 0.00084 | -0.00370 | -0.00906 | -0.01442 | -0.01896 | -0.02200 | -0.02306 |
| 9.4154 | 0.00606 | 0.00505 | 0.00220 | -0.00208 | -0.00712 | -0.01216 | -0.01643 | -0.01928 | -0.02029 |
| 10.1397 | 0.00671 | 0.00576 | 0.00308 | -0.00094 | -0.00568 | -0.01042 | -0.01444 | -0.01712 | -0.01807 |
| 10.8639 | 0.00800 | 0.00713 | 0.00462 | 0.00087 | -0.00355 | -0.00797 | -0.01172 | -0.01422 | -0.01510 |
| 11.5882 | 0.00871 | 0.00790 | 0.00559 | 0.00213 | -0.00196 | -0.00604 | -0.00950 | -0.01181 | -0.01263 |
| 12.3124 | 0.01038 | 0.00964 | 0.00753 | 0.00436 | 0.00063 | -0.00311 | -0.00627 | -0.00839 | -0.00913 |
| 13.0367 | 0.01105 | 0.01039 | 0.00851 | 0.00569 | 0.00236 | -0.00096 | -0.00378 | -0.00566 | -0.00632 |
| 13.7610 | 0.01238 | 0.01180 | 0.01014 | 0.00765 | 0.00472 | 0.00179 | -0.00070 | -0.00236 | -0.00294 |
| 14.5023 | 0.01363 | 0.01315 | 0.01177 | 0.00970 | 0.00726 | 0.00481 | 0.00275 | 0.00136 | 0.00088 |
| 15.2436 | 0.01910 | 0.01872 | 0.01765 | 0.01605 | 0.01417 | 0.01228 | 0.01068 | 0.00961 | 0.00924 |
| 15.9849 | 0.01971 | 0.01951 | 0.01894 | 0.01808 | 0.01707 | 0.01606 | 0.01520 | 0.01463 | 0.01443 |
| 16.7262 | 0.01345 | 0.01337 | 0.01315 | 0.01283 | 0.01244 | 0.01206 | 0.01173 | 0.01151 | 0.01144 |
| 17.4675 | 0.01081 | 0.01081 | 0.01084 | 0.01087 | 0.01091 | 0.01095 | 0.01098 | 0.01100 | 0.01101 |
| 18.2088 | 0.00830 | 0.00836 | 0.00853 | 0.00878 | 0.00907 | 0.00937 | 0.00962 | 0.00979 | 0.00985 |
| 18.9501 | 0.00667 | 0.00676 | 0.00700 | 0.00737 | 0.00780 | 0.00824 | 0.00861 | 0.00885 | 0.00894 |
| 19.6914 | 0.00544 | 0.00554 | 0.00582 | 0.00623 | 0.00672 | 0.00720 | 0.00762 | 0.00789 | 0.00799 |
| 20.4327 | 0.00459 | 0.00468 | 0.00495 | 0.00536 | 0.00585 | 0.00633 | 0.00674 | 0.00701 | 0.00711 |
| 21.1740 | 0.00396 | 0.00405 | 0.00430 | 0.00468 | 0.00512 | 0.00557 | 0.00594 | 0.00620 | 0.00629 |
| 21.9153 | 0.00350 | 0.00358 | 0.00380 | 0.00413 | 0.00452 | 0.00491 | 0.00524 | 0.00546 | 0.00554 |
| 22.6566 | 0.00316 | 0.00322 | 0.00341 | 0.00369 | 0.00402 | 0.00434 | 0.00462 | 0.00481 | 0.00487 |
| 23.3979 | 0.00289 | 0.00294 | 0.00309 | 0.00332 | 0.00359 | 0.00386 | 0.00408 | 0.00423 | 0.00429 |
| 24.1392 | 0.00267 | 0.00271 | 0.00283 | 0.00301 | 0.00322 | 0.00344 | 0.00362 | 0.00374 | 0.00378 |
| 24.8805 | 0.00248 | 0.00251 | 0.00260 | 0.00274 | 0.00291 | 0.00307 | 0.00322 | 0.00331 | 0.00334 |
| 25.6218 | 0.00231 | 0.00233 | 0.00240 | 0.00251 | 0.00264 | 0.00276 | 0.00287 | 0.00294 | 0.00297 |
| 26.3631 | 0.00215 | 0.00217 | 0.00223 | 0.00231 | 0.00240 | 0.00250 | 0.00258 | 0.00263 | 0.00265 |
| 27.1044 | 0.00201 | 0.00202 | 0.00206 | 0.00212 | 0.00219 | 0.00226 | 0.00232 | 0.00236 | 0.00238 |
| 27.8457 | 0.00189 | 0.00190 | 0.00193 | 0.00198 | 0.00203 | 0.00209 | 0.00213 | 0.00216 | 0.00217 |
| 28.5870 | 0.00169 | 0.00170 | 0.00172 | 0.00176 | 0.00180 | 0.00184 | 0.00187 | 0.00190 | 0.00190 |
| 29.3283 | 0.00186 | 0.00186 | 0.00188 | 0.00190 | 0.00194 | 0.00197 | 0.00199 | 0.00201 | 0.00202 |
| 30.0696 | 0.00441 | 0.00442 | 0.00447 | 0.00455 | 0.00464 | 0.00473 | 0.00480 | 0.00485 | 0.00487 |
| 30.8109 | 0.00816 | 0.00823 | 0.00843 | 0.00873 | 0.00909 | 0.00944 | 0.00974 | 0.00994 | 0.01001 |
| 31.5522 | 0.01013 | 0.01029 | 0.01075 | 0.01144 | 0.01226 | 0.01307 | 0.01376 | 0.01422 | 0.01438 |
| 32.2935 | 0.01030 | 0.01057 | 0.01136 | 0.01253 | 0.01390 | 0.01528 | 0.01645 | 0.01724 | 0.01751 |
| 33.0348 | 0.01130 | 0.01170 | 0.01285 | 0.01456 | 0.01658 | 0.01860 | 0.02031 | 0.02145 | 0.02185 |
| 33.7761 | 0.01218 | 0.01274 | 0.01434 | 0.01673 | 0.01955 | 0.02237 | 0.02476 | 0.02636 | 0.02692 |
| 34.5174 | 0.00788 | 0.00861 | 0.01067 | 0.01375 | 0.01738 | 0.02101 | 0.02409 | 0.02615 | 0.02687 |
| 35.2587 | -0.00143 | -0.00062 | 0.00167 | 0.00510 | 0.00915 | 0.01320 | 0.01663 | 0.01892 | 0.01973 |

RACIAL(VR)

| THETA(DEG.) X | 0.0000 | 22.5000 | 45.0000 | 67.5000 | 90.0000 | 112.5000 | 135.0000 | 157.5000 | 180.0000 |
|---|---|---|---|---|---|---|---|---|---|
| 0.0000 | 0.06980 | 0.07733 | 0.09878 | 0.13087 | 0.16873 | 0.20658 | 0.23867 | 0.26012 | 0.26765 |
| 0.7243 | 0.07603 | 0.08356 | 0.10501 | 0.13710 | 0.17496 | 0.21281 | 0.24490 | 0.26635 | 0.27388 |
| 1.4485 | 0.04391 | 0.05152 | 0.07319 | 0.10561 | 0.14387 | 0.18212 | 0.21455 | 0.23621 | 0.24382 |
| 2.1728 | 0.00905 | 0.01681 | 0.03893 | 0.07202 | 0.11105 | 0.15009 | 0.18318 | 0.20529 | 0.21305 |
| 2.8970 | 0.00551 | 0.01329 | 0.03545 | 0.06861 | 0.10772 | 0.14683 | 0.17999 | 0.20214 | 0.20992 |
| 3.6213 | -0.00626 | 0.00156 | 0.02383 | 0.05715 | 0.09646 | 0.13576 | 0.16909 | 0.19135 | 0.19917 |
| 4.3456 | -0.01079 | -0.00296 | 0.01934 | 0.05272 | 0.09209 | 0.13146 | 0.16484 | 0.18714 | 0.19498 |
| 5.0698 | -0.01777 | -0.00991 | 0.01244 | 0.04590 | 0.08538 | 0.12485 | 0.15831 | 0.18066 | 0.18852 |
| 5.7941 | -0.02206 | -0.01420 | 0.00819 | 0.04169 | 0.08122 | 0.12074 | 0.15425 | 0.17664 | 0.18450 |
| 6.5184 | -0.02759 | -0.01972 | 0.00271 | 0.03628 | 0.07587 | 0.11546 | 0.14902 | 0.17145 | 0.17933 |
| 7.2426 | -0.03165 | -0.02377 | -0.00131 | 0.03229 | 0.07193 | 0.11157 | 0.14517 | 0.16763 | 0.17551 |
| 7.9669 | -0.03667 | -0.02877 | -0.00629 | 0.02736 | 0.06705 | 0.10674 | 0.14039 | 0.16287 | 0.17077 |
| 8.6911 | -0.04058 | -0.03268 | -0.01017 | 0.02351 | 0.06324 | 0.10298 | 0.13666 | 0.15917 | 0.16707 |
| 9.4154 | -0.04529 | -0.03738 | -0.01485 | 0.01887 | 0.05865 | 0.09843 | 0.13215 | 0.15468 | 0.16259 |
| 10.1397 | -0.04935 | -0.04143 | -0.01888 | 0.01488 | 0.05469 | 0.09450 | 0.12826 | 0.15081 | 0.15873 |
| 10.8639 | -0.05448 | -0.04655 | -0.02398 | 0.00981 | 0.04966 | 0.08952 | 0.12330 | 0.14588 | 0.15381 |
| 11.5882 | -0.05896 | -0.05103 | -0.02843 | 0.00538 | 0.04527 | 0.08516 | 0.11898 | 0.14157 | 0.14951 |
| 12.3124 | -0.06507 | -0.05713 | -0.03451 | -0.00066 | 0.03927 | 0.07920 | 0.11305 | 0.13566 | 0.14361 |
| 13.0367 | -0.07015 | -0.06220 | -0.03956 | -0.00569 | 0.03427 | 0.07423 | 0.10811 | 0.13075 | 0.13870 |
| 13.7610 | -0.07630 | -0.06835 | -0.04569 | -0.01179 | 0.02820 | 0.06819 | 0.10210 | 0.12475 | 0.13271 |
| 14.5023 | -0.08310 | -0.07513 | -0.05246 | -0.01853 | 0.02149 | 0.06151 | 0.09544 | 0.11811 | 0.12607 |
| 15.2436 | -0.09731 | -0.08934 | -0.06665 | -0.03268 | 0.00738 | 0.04744 | 0.08140 | 0.10409 | 0.11206 |
| 15.9849 | -0.10671 | -0.09874 | -0.07603 | -0.04206 | -0.00198 | 0.03809 | 0.07207 | 0.09477 | 0.10274 |
| 16.7262 | -0.10419 | -0.09622 | -0.07352 | -0.03955 | 0.00053 | 0.04060 | 0.07458 | 0.09728 | 0.10525 |
| 17.4675 | -0.10486 | -0.09689 | -0.07419 | -0.04022 | -0.00014 | 0.03993 | 0.07391 | 0.09661 | 0.10458 |
| 18.2088 | -0.10468 | -0.09671 | -0.07401 | -0.04004 | 0.00004 | 0.04011 | 0.07409 | 0.09679 | 0.10476 |
| 18.9501 | -0.10473 | -0.09676 | -0.07406 | -0.04008 | -0.00001 | 0.04006 | 0.07404 | 0.09674 | 0.10471 |
| 19.6914 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | 0.00000 | 0.04008 | 0.07405 | 0.09675 | 0.10472 |
| 20.4327 | -0.10472 | -0.09675 | -0.07405 | -0.04008 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 21.1740 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | 0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 21.9153 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 22.6566 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 23.3979 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 24.1392 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 24.8805 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 25.6218 | -0.10472 | -0.09675 | -0.07405 | -0.04007 | 0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 26.3631 | -0.10472 | -0.09675 | -0.07405 | -0.04008 | -0.00000 | 0.04007 | 0.07405 | 0.09675 | 0.10472 |
| 27.1044 | -0.10471 | -0.09674 | -0.07404 | -0.04007 | 0.00001 | 0.04008 | 0.07405 | 0.09675 | 0.10473 |
| 27.8457 | -0.10474 | -0.09677 | -0.07407 | -0.04010 | -0.00002 | 0.04005 | 0.07402 | 0.09672 | 0.10470 |
| 28.5870 | -0.10463 | -0.09666 | -0.07396 | -0.03998 | 0.00009 | 0.04017 | 0.07414 | 0.09684 | 0.10481 |
| 29.3283 | -0.10506 | -0.09709 | -0.07439 | -0.04041 | -0.00034 | 0.03973 | 0.07371 | 0.09641 | 0.10438 |
| 30.0696 | -0.10972 | -0.10174 | -0.07904 | -0.04507 | -0.00500 | 0.03508 | 0.06905 | 0.09175 | 0.09972 |
| 30.8109 | -0.11810 | -0.11013 | -0.08743 | -0.05347 | -0.01340 | 0.02667 | 0.06064 | 0.08334 | 0.09131 |
| 31.5522 | -0.12578 | -0.11781 | -0.09512 | -0.06116 | -0.02110 | 0.01895 | 0.05291 | 0.07560 | 0.08357 |
| 32.2935 | -0.13186 | -0.12390 | -0.10122 | -0.06728 | -0.02724 | 0.01280 | 0.04674 | 0.06942 | 0.07738 |
| 33.0348 | -0.14009 | -0.13214 | -0.10948 | -0.07556 | -0.03556 | 0.00444 | 0.03836 | 0.06102 | 0.06898 |
| 33.7761 | -0.14989 | -0.14194 | -0.11931 | -0.08545 | -0.04550 | -0.00555 | 0.02832 | 0.05094 | 0.05889 |
| 34.5174 | -0.15255 | -0.14461 | -0.12201 | -0.08818 | -0.04828 | -0.00838 | 0.02545 | 0.04805 | 0.05599 |
| 35.2587 | -0.14441 | -0.13647 | -0.11387 | -0.08003 | -0.04012 | -0.00020 | 0.03363 | 0.05624 | 0.06418 |

TANGENTIAL(VT)

| THETA(DEG.) X | 0.0000 | 22.5000 | 45.0000 | 67.5000 | 90.0000 | 112.5000 | 135.0000 | 157.5000 | 180.0000 |
|---|---|---|---|---|---|---|---|---|---|
| 0.0000 | -0.00000 | -0.02800 | -0.05174 | -0.06760 | -0.07316 | -0.06760 | -0.05174 | -0.02800 | -0.00000 |
| 0.7243 | -0.00000 | -0.02800 | -0.05174 | -0.06760 | -0.07316 | -0.06760 | -0.05174 | -0.02800 | -0.00000 |
| 1.4485 | -0.00000 | -0.02816 | -0.05202 | -0.06797 | -0.07357 | -0.06797 | -0.05202 | -0.02816 | -0.00000 |
| 2.1728 | -0.00000 | -0.03396 | -0.06275 | -0.08198 | -0.08874 | -0.08198 | -0.06275 | -0.03396 | -0.00000 |
| 2.8970 | -0.00000 | -0.03384 | -0.06252 | -0.08168 | -0.08842 | -0.08168 | -0.06252 | -0.03384 | -0.00000 |
| 3.6213 | -0.00000 | -0.03527 | -0.06518 | -0.08516 | -0.09218 | -0.08516 | -0.06518 | -0.03527 | -0.00000 |
| 4.3456 | -0.00000 | -0.03545 | -0.06550 | -0.08559 | -0.09264 | -0.08559 | -0.06550 | -0.03545 | -0.00000 |
| 5.0698 | -0.00000 | -0.03615 | -0.06680 | -0.08728 | -0.09447 | -0.08728 | -0.06680 | -0.03615 | -0.00000 |
| 5.7941 | -0.00000 | -0.03641 | -0.06728 | -0.08790 | -0.09515 | -0.08790 | -0.06728 | -0.03641 | -0.00000 |
| 6.5184 | -0.00000 | -0.03690 | -0.06818 | -0.08909 | -0.09643 | -0.08909 | -0.06818 | -0.03690 | -0.00000 |
| 7.2426 | -0.00000 | -0.03718 | -0.06870 | -0.08976 | -0.09716 | -0.08976 | -0.06870 | -0.03718 | -0.00000 |
| 7.9669 | -0.00000 | -0.03758 | -0.06944 | -0.09073 | -0.09820 | -0.09073 | -0.06944 | -0.03758 | -0.00000 |
| 8.6911 | -0.00000 | -0.03788 | -0.06998 | -0.09144 | -0.09897 | -0.09144 | -0.06998 | -0.03788 | -0.00000 |
| 9.4154 | -0.00000 | -0.03822 | -0.07063 | -0.09228 | -0.09988 | -0.09228 | -0.07063 | -0.03822 | -0.00000 |
| 10.1397 | -0.00000 | -0.03853 | -0.07119 | -0.09301 | -0.10067 | -0.09301 | -0.07119 | -0.03853 | -0.00000 |
| 10.8639 | -0.00000 | -0.03886 | -0.07180 | -0.09382 | -0.10155 | -0.09382 | -0.07180 | -0.03886 | -0.00000 |
| 11.5882 | -0.00000 | -0.03920 | -0.07244 | -0.09465 | -0.10245 | -0.09465 | -0.07244 | -0.03920 | -0.00000 |
| 12.3124 | -0.00000 | -0.03956 | -0.07309 | -0.09550 | -0.10337 | -0.09550 | -0.07309 | -0.03956 | -0.00000 |
| 13.0367 | -0.00000 | -0.03998 | -0.07387 | -0.09651 | -0.10447 | -0.09651 | -0.07387 | -0.03998 | -0.00000 |
| 13.7610 | -0.00000 | -0.04034 | -0.07454 | -0.09739 | -0.10542 | -0.09739 | -0.07454 | -0.04034 | -0.00000 |
| 14.5023 | -0.00000 | -0.04082 | -0.07543 | -0.09856 | -0.10668 | -0.09856 | -0.07543 | -0.04082 | -0.00000 |
| 15.2436 | -0.00000 | -0.04136 | -0.07643 | -0.09986 | -0.10809 | -0.09986 | -0.07643 | -0.04136 | -0.00000 |
| 15.9849 | -0.00000 | -0.04229 | -0.07814 | -0.10210 | -0.11051 | -0.10210 | -0.07814 | -0.04229 | -0.00000 |
| 16.7262 | -0.00000 | -0.04270 | -0.07891 | -0.10310 | -0.11159 | -0.10310 | -0.07891 | -0.04270 | -0.00000 |
| 17.4675 | -0.00000 | -0.04281 | -0.07910 | -0.10335 | -0.11186 | -0.10335 | -0.07910 | -0.04281 | -0.00000 |
| 18.2088 | -0.00000 | -0.04269 | -0.07889 | -0.10307 | -0.11156 | -0.10307 | -0.07889 | -0.04269 | -0.00000 |
| 18.9501 | -0.00000 | -0.04246 | -0.07845 | -0.10251 | -0.11095 | -0.10251 | -0.07845 | -0.04246 | -0.00000 |
| 19.6914 | -0.00000 | -0.04217 | -0.07791 | -0.10180 | -0.11019 | -0.10180 | -0.07791 | -0.04217 | -0.00000 |
| 20.4327 | -0.00000 | -0.04186 | -0.07735 | -0.10106 | -0.10939 | -0.10106 | -0.07735 | -0.04186 | -0.00000 |
| 21.1740 | -0.00000 | -0.04157 | -0.07681 | -0.10036 | -0.10862 | -0.10036 | -0.07681 | -0.04157 | -0.00000 |
| 21.9153 | -0.00000 | -0.04131 | -0.07632 | -0.09972 | -0.10794 | -0.09972 | -0.07632 | -0.04131 | -0.00000 |
| 22.6566 | -0.00000 | -0.04108 | -0.07591 | -0.09918 | -0.10735 | -0.09918 | -0.07591 | -0.04108 | -0.00000 |
| 23.3979 | -0.00000 | -0.04090 | -0.07556 | -0.09873 | -0.10686 | -0.09873 | -0.07556 | -0.04090 | -0.00000 |
| 24.1392 | -0.00000 | -0.04075 | -0.07529 | -0.09837 | -0.10647 | -0.09837 | -0.07529 | -0.04075 | -0.00000 |
| 24.8805 | -0.00000 | -0.04063 | -0.07507 | -0.09808 | -0.10616 | -0.09808 | -0.07507 | -0.04063 | -0.00000 |
| 25.6218 | -0.00000 | -0.04054 | -0.07490 | -0.09786 | -0.10592 | -0.09786 | -0.07490 | -0.04054 | -0.00000 |
| 26.3631 | -0.00000 | -0.04047 | -0.07477 | -0.09770 | -0.10574 | -0.09770 | -0.07477 | -0.04047 | -0.00000 |
| 27.1044 | -0.00000 | -0.04041 | -0.07468 | -0.09757 | -0.10561 | -0.09757 | -0.07468 | -0.04041 | -0.00000 |
| 27.8457 | -0.00000 | -0.04038 | -0.07461 | -0.09748 | -0.10551 | -0.09748 | -0.07461 | -0.04038 | -0.00000 |
| 28.5870 | -0.00000 | -0.04035 | -0.07455 | -0.09741 | -0.10543 | -0.09741 | -0.07455 | -0.04035 | -0.00000 |
| 29.3283 | -0.00000 | -0.04032 | -0.07451 | -0.09735 | -0.10537 | -0.09735 | -0.07451 | -0.04032 | -0.00000 |
| 30.0696 | -0.00000 | -0.04040 | -0.07464 | -0.09753 | -0.10556 | -0.09753 | -0.07464 | -0.04040 | -0.00000 |
| 30.8109 | -0.00000 | -0.04073 | -0.07526 | -0.09833 | -0.10643 | -0.09833 | -0.07526 | -0.04073 | -0.00000 |
| 31.5522 | -0.00000 | -0.04125 | -0.07623 | -0.09960 | -0.10780 | -0.09960 | -0.07623 | -0.04125 | -0.00000 |
| 32.2935 | -0.00000 | -0.04181 | -0.07726 | -0.10094 | -0.10926 | -0.10094 | -0.07726 | -0.04181 | -0.00000 |
| 33.0348 | -0.00000 | -0.04234 | -0.07823 | -0.10222 | -0.11064 | -0.10222 | -0.07823 | -0.04234 | -0.00000 |
| 33.7761 | -0.00000 | -0.04294 | -0.07935 | -0.10367 | -0.11222 | -0.10367 | -0.07935 | -0.04294 | -0.00000 |
| 34.5174 | -0.00000 | -0.04341 | -0.08020 | -0.10479 | -0.11342 | -0.10479 | -0.08020 | -0.04341 | -0.00000 |
| 35.2587 | -0.00000 | -0.04328 | -0.07996 | -0.10448 | -0.11308 | -0.10448 | -0.07996 | -0.04328 | -0.00000 |

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.00162 | 0.00033 | 0.00319 | 0.00195 | 0.00179 | 0.00086 | -0.00018 | 0.00012 | -0.00147 | -0.00103 |
| 2 | 0.00068 | 0.00436 | -0.00288 | 0.00126 | 0.00064 | -0.00021 | 0.00003 | -0.00133 | -0.00112 | -0.00249 |
| 3 | 0.00668 | 0.00803 | 0.01687 | 0.00050 | -0.00035 | -0.00025 | -0.00162 | -0.00117 | -0.00266 | -0.00144 |
| 4 | 0.01016 | 0.00300 | 0.00032 | -0.00016 | -0.00074 | -0.00242 | -0.00129 | -0.00277 | -0.00135 | -0.00108 |
| 5 | 0.00599 | 0.00136 | 0.00048 | 0.00014 | -0.00340 | -0.00137 | -0.00320 | -0.00152 | -0.00022 | -0.00010 |
| 6 | 0.00375 | 0.00145 | -0.00081 | -0.00025 | -0.00154 | -0.00455 | -0.00167 | -0.00010 | -0.00057 | -0.00043 |
| 7 | 0.00377 | -0.00045 | 0.00004 | -0.00183 | -0.00132 | -0.00190 | -0.00049 | -0.00051 | 0.00049 | 0.00029 |
| 8 | 0.00073 | 0.00037 | -0.00257 | -0.00141 | -0.00223 | -0.00043 | -0.00056 | 0.00046 | 0.00030 | 0.00049 |
| 9 | 0.00132 | -0.00295 | -0.00115 | -0.00266 | -0.00029 | -0.00063 | 0.00070 | 0.00031 | 0.00002 | 0.00018 |
| 10 | -0.00389 | -0.00248 | -0.00159 | 0.00019 | -0.00079 | 0.00097 | 0.00030 | 0.00001 | 0.00004 | 0.00019 |

TRANSVERSE(V)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.00452 | 0.00007 | -0.00488 | -0.00352 | -0.00343 | -0.00199 | -0.00031 | -0.00064 | 0.00189 | 0.00139 |
| 2 | 0.00072 | -0.00673 | 0.00337 | -0.00292 | -0.00189 | -0.00047 | -0.00063 | 0.00159 | 0.00144 | 0.00363 |
| 3 | -0.01002 | -0.01350 | -0.02684 | -0.00210 | -0.00056 | -0.00042 | 0.00188 | 0.00144 | 0.00385 | 0.00213 |
| 4 | -0.01793 | -0.00745 | -0.00264 | -0.00136 | -0.00001 | 0.00287 | 0.00151 | 0.00395 | 0.00201 | 0.00164 |
| 5 | -0.01586 | -0.00620 | -0.00337 | -0.00189 | 0.00397 | 0.00145 | 0.00449 | 0.00223 | 0.00035 | 0.00019 |
| 6 | -0.01600 | -0.00743 | -0.00166 | -0.00135 | 0.00140 | 0.00633 | 0.00240 | 0.00016 | 0.00091 | 0.00069 |
| 7 | -0.01924 | -0.00529 | -0.00306 | 0.00127 | 0.00123 | 0.00264 | 0.00072 | 0.00083 | -0.00069 | -0.00037 |
| 8 | -0.01760 | -0.00690 | 0.00113 | 0.00089 | 0.00294 | 0.00056 | 0.00087 | -0.00066 | -0.00042 | -0.00070 |
| 9 | -0.02023 | -0.00169 | -0.00059 | 0.00317 | 0.00023 | 0.00095 | -0.00101 | -0.00044 | -0.00001 | -0.00024 |
| 10 | -0.01337 | -0.00168 | 0.00062 | -0.00076 | 0.00109 | -0.00144 | -0.00042 | -0.00001 | -0.00004 | -0.00027 |

VERTICAL(W)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.02600 | 0.01023 | 0.00683 | 0.00500 | 0.00497 | 0.00480 | 0.00443 | 0.00422 | 0.00386 | 0.00348 |
| 2 | 0.04572 | 0.01432 | 0.00763 | 0.00682 | 0.00627 | 0.00554 | 0.00511 | 0.00457 | 0.00424 | 0.00351 |
| 3 | 0.06508 | 0.01802 | 0.01118 | 0.00880 | 0.00727 | 0.00638 | 0.00556 | 0.00503 | 0.00437 | 0.00368 |
| 4 | 0.08566 | 0.02214 | 0.01399 | 0.01025 | 0.00844 | 0.00693 | 0.00609 | 0.00522 | 0.00472 | 0.00385 |
| 5 | 0.10355 | 0.02870 | 0.01633 | 0.01187 | 0.00936 | 0.00766 | 0.00639 | 0.00557 | 0.00487 | 0.00393 |
| 6 | 0.12660 | 0.03353 | 0.01868 | 0.01328 | 0.01007 | 0.00813 | 0.00680 | 0.00586 | 0.00487 | 0.00391 |
| 7 | 0.14709 | 0.03801 | 0.02102 | 0.01421 | 0.01096 | 0.00857 | 0.00718 | 0.00584 | 0.00500 | 0.00402 |
| 8 | 0.16568 | 0.04265 | 0.02247 | 0.01548 | 0.01130 | 0.00907 | 0.00720 | 0.00597 | 0.00502 | 0.00400 |
| 9 | 0.18273 | 0.04543 | 0.02449 | 0.01594 | 0.01201 | 0.00916 | 0.00737 | 0.00605 | 0.00488 | 0.00400 |
| 10 | 0.19324 | 0.04899 | 0.02524 | 0.01706 | 0.01215 | 0.00942 | 0.00747 | 0.00591 | 0.00484 | 0.00401 |

VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LINE SOURCES AND DOUBLETS

AXIAL(U)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1C |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.01160 | 0.01127 | 0.00687 | 0.00506 | 0.00392 | 0.00304 | 0.00243 | 0.00199 | 0.00165 | 0.00136 |
| 2 | 0.01424 | 0.00795 | 0.00583 | 0.00439 | 0.00335 | 0.00265 | 0.00214 | 0.00177 | 0.00148 | 0.00127 |
| 3 | 0.00970 | 0.00699 | 0.00502 | 0.00374 | 0.00291 | 0.00233 | 0.00190 | 0.00158 | 0.00134 | 0.00119 |
| 4 | 0.00877 | 0.00588 | 0.00425 | 0.00324 | 0.00255 | 0.00206 | 0.00170 | 0.00143 | 0.00122 | 0.00112 |
| 5 | 0.00718 | 0.00496 | 0.00368 | 0.00283 | 0.00226 | 0.00184 | 0.00153 | 0.00130 | 0.00111 | 0.00105 |
| 6 | 0.00604 | 0.00427 | 0.00320 | 0.00250 | 0.00201 | 0.00165 | 0.00139 | 0.00118 | 0.00102 | 0.00100 |
| 7 | 0.00516 | 0.00369 | 0.00280 | 0.00221 | 0.00180 | 0.00150 | 0.00126 | 0.00109 | 0.00097 | 0.00093 |
| 8 | 0.00442 | 0.00321 | 0.00247 | 0.00198 | 0.00162 | 0.00136 | 0.00116 | 0.00101 | 0.00191 | 0.00090 |
| 9 | 0.00382 | 0.00282 | 0.00220 | 0.00178 | 0.00147 | 0.00125 | 0.00106 | 0.00178 | 0.00403 | 0.00117 |
| 10 | 0.00333 | 0.00250 | 0.00197 | 0.00161 | 0.00135 | 0.00112 | 0.00167 | 0.00397 | 0.00543 | 0.00220 |

TRANSVERSE(V)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.C0943 | -0.00701 | -0.00546 | -0.00487 | -0.00423 | -0.00351 | -0.00296 | -0.00253 | -0.00219 | -0.00185 |
| 2 | 0.00100 | -0.00408 | -0.00502 | -0.00454 | -0.00380 | -0.00322 | -0.00274 | -0.00236 | -0.00205 | -0.00178 |
| 3 | 0.00407 | -0.00436 | -0.00473 | -0.00412 | -0.00352 | -0.00300 | -0.00257 | -0.00223 | -0.00195 | -C.00172 |
| 4 | 0.00249 | -0.00411 | -0.00438 | -0.00388 | -0.00332 | -0.00283 | -0.00244 | -0.00212 | -0.00186 | -0.00167 |
| 5 | 0.00235 | -0.00396 | -0.00421 | -0.00370 | -0.00316 | -0.00271 | -0.00234 | -0.00204 | -0.00179 | -0.00162 |
| 6 | 0.00187 | -0.00399 | -0.00410 | -0.00358 | -0.00305 | -0.00261 | -0.00226 | -0.00197 | -0.00174 | -0.C0159 |
| 7 | 0.00128 | -0.00404 | -0.00404 | -0.00350 | -0.00298 | -0.00255 | -0.00220 | -0.00193 | -0.00175 | -0.00153 |
| 8 | 0.00072 | -0.00414 | -0.00403 | -0.00345 | -0.00293 | -0.00250 | -0.00217 | -0.00191 | -0.00325 | -0.00153 |
| 9 | 0.00014 | -0.00428 | -0.00405 | -0.00344 | -0.00290 | -0.00248 | -0.00213 | -0.00317 | -0.00662 | -0.00198 |
| 10 | -0.00045 | -0.00445 | -0.00410 | -0.00344 | -0.00290 | -0.00242 | -0.00316 | -0.00665 | -0.00900 | -0.00358 |

VERTICAL(W)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.06109 | 0.02978 | 0.01711 | 0.01098 | 0.00755 | 0.00546 | 0.00412 | 0.00321 | 0.00258 | 0.00199 |
| 2 | 0.06317 | 0.02928 | 0.01671 | 0.01066 | 0.00731 | 0.00530 | 0.00401 | 0.00314 | 0.00253 | 0.C0197 |
| 3 | 0.06277 | 0.02888 | 0.01630 | 0.01035 | 0.00712 | 0.00518 | 0.00393 | 0.00309 | 0.00250 | 0.00195 |
| 4 | 0.06264 | 0.02835 | 0.01591 | 0.01011 | 0.00696 | 0.00508 | 0.00387 | 0.00305 | 0.00247 | 0.00194 |
| 5 | 0.06200 | 0.02783 | 0.01559 | 0.00991 | 0.00684 | 0.00501 | 0.00383 | 0.00302 | 0.00245 | 0.00192 |
| 6 | 0.06133 | 0.02739 | 0.01533 | 0.00976 | 0.00675 | 0.00495 | 0.00379 | 0.00300 | 0.00244 | 0.0C191 |
| 7 | 0.06071 | 0.02703 | 0.01513 | 0.00964 | 0.00668 | 0.00491 | 0.00377 | 0.00298 | 0.00242 | 0.00190 |
| 8 | 0.06017 | 0.02675 | 0.01498 | 0.00956 | 0.00663 | 0.00488 | 0.00374 | 0.00297 | 0.00246 | 0.00190 |
| 9 | 0.05976 | 0.02654 | 0.01487 | 0.00950 | 0.00660 | 0.00486 | 0.00373 | 0.00300 | 0.00253 | 0.CC190 |
| 10 | 0.05946 | 0.02640 | 0.01480 | 0.00946 | 0.00657 | 0.00484 | 0.00375 | 0.00309 | 0.00253 | 0.00194 |

VELOCITY COMPONENTS ON WING PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.08732 | 0.10456 | 0.11606 | 0.12644 | 0.13618 | 0.14457 | 0.15156 | 0.15736 | 0.16407 | 0.26144 |
| 2 | 0.05305 | 0.06110 | 0.06789 | 0.07433 | 0.07938 | 0.08339 | 0.08656 | 0.08917 | 0.09606 | 0.14600 |
| 3 | 0.04340 | 0.04878 | 0.05427 | 0.05829 | 0.06115 | 0.06323 | 0.06459 | 0.06645 | 0.08029 | 0.10417 |
| 4 | 0.03934 | 0.04306 | 0.04699 | 0.04930 | 0.05076 | 0.05134 | 0.05181 | 0.05411 | 0.06970 | 0.07089 |
| 5 | 0.03654 | 0.03994 | 0.04170 | 0.04271 | 0.04275 | 0.04248 | 0.04273 | 0.04651 | 0.05846 | 0.05362 |
| 6 | 0.03577 | 0.03666 | 0.03711 | 0.03667 | 0.03569 | 0.03518 | 0.03573 | 0.04122 | 0.04655 | 0.04124 |
| 7 | 0.03397 | 0.03334 | 0.03223 | 0.03049 | 0.02937 | 0.02891 | 0.02994 | 0.03709 | 0.03512 | 0.03063 |
| 8 | 0.03129 | 0.02890 | 0.02609 | 0.02425 | 0.02322 | 0.02304 | 0.02463 | 0.03113 | 0.02443 | 0.02002 |
| 9 | 0.02686 | 0.02207 | 0.01923 | 0.01770 | 0.01712 | 0.01701 | 0.01929 | 0.02306 | 0.01460 | 0.00915 |
| 10 | 0.01781 | 0.01315 | 0.01120 | 0.01059 | 0.01008 | 0.01026 | 0.01238 | 0.01301 | 0.00618 | -0.00418 |

TRANSVERSE(V)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.23989 | -0.28726 | -0.31885 | -0.34737 | -0.37414 | -0.39718 | -0.41638 | -0.43232 | -0.45077 | -0.71827 |
| 2 | -0.14575 | -0.16786 | -0.18652 | -0.20420 | -0.21809 | -0.22911 | -0.23781 | -0.24497 | -0.26392 | -0.45010 |
| 3 | -0.11923 | -0.13401 | -0.14909 | -0.16013 | -0.16799 | -0.17372 | -0.17744 | -0.18255 | -0.22059 | -0.37067 |
| 4 | -0.10808 | -0.11830 | -0.12910 | -0.13545 | -0.13944 | -0.14104 | -0.14235 | -0.14864 | -0.19149 | -0.32161 |
| 5 | -0.10039 | -0.10973 | -0.11455 | -0.11734 | -0.11746 | -0.11669 | -0.11738 | -0.12778 | -0.16061 | -0.30347 |
| 6 | -0.09827 | -0.10073 | -0.10196 | -0.10073 | -0.09806 | -0.09666 | -0.09815 | -0.11325 | -0.12787 | -0.29573 |
| 7 | -0.09333 | -0.09159 | -0.08854 | -0.08375 | -0.08067 | -0.07942 | -0.08226 | -0.10190 | -0.09647 | -0.29359 |
| 8 | -0.08597 | -0.07941 | -0.07169 | -0.06662 | -0.06378 | -0.06329 | -0.06765 | -0.08552 | -0.06710 | -0.29596 |
| 9 | -0.07380 | -0.06062 | -0.05282 | -0.04861 | -0.04703 | -0.04672 | -0.05297 | -0.06335 | -0.04011 | -0.30300 |
| 10 | -0.04893 | -0.03611 | -0.03076 | -0.02908 | -0.02769 | -0.02819 | -0.03399 | -0.03573 | -0.01696 | -0.31729 |

VERTICAL(W)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.19182 | -0.14473 | -0.12866 | -0.12070 | -0.11725 | -0.11498 | -0.11327 | -0.11215 | -0.11116 | -0.11018 |
| 2 | -0.21361 | -0.14832 | -0.12906 | -0.12220 | -0.11830 | -0.11557 | -0.11385 | -0.11243 | -0.11150 | -0.11019 |
| 3 | -0.23257 | -0.15162 | -0.13220 | -0.12387 | -0.11910 | -0.11628 | -0.11421 | -0.11284 | -0.11159 | -0.11035 |
| 4 | -0.25302 | -0.15521 | -0.13462 | -0.12508 | -0.12012 | -0.11673 | -0.11469 | -0.11300 | -0.11191 | -0.11050 |
| 5 | -0.27027 | -0.16125 | -0.13664 | -0.12650 | -0.12092 | -0.11738 | -0.11494 | -0.11331 | -0.11204 | -0.11057 |
| 6 | -0.29265 | -0.16564 | -0.13873 | -0.12775 | -0.12154 | -0.11780 | -0.11531 | -0.11358 | -0.11202 | -0.11054 |
| 7 | -0.31251 | -0.16975 | -0.14087 | -0.12857 | -0.12236 | -0.11820 | -0.11567 | -0.11354 | -0.11214 | -0.11065 |
| 8 | -0.33057 | -0.17412 | -0.14216 | -0.12976 | -0.12266 | -0.11867 | -0.11566 | -0.11366 | -0.11219 | -0.11061 |
| 9 | -0.34721 | -0.17669 | -0.14408 | -0.13015 | -0.12333 | -0.11873 | -0.11582 | -0.11377 | -0.11213 | -0.11062 |
| 10 | -0.35742 | -0.18011 | -0.14475 | -0.13124 | -0.12344 | -0.11897 | -0.11594 | -0.11371 | -0.11209 | -0.11066 |

VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOURCES

AXIAL(U)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.01440 | -0.00890 | -0.00478 | -0.00453 | 0.00006 | 0.00447 | 0.00827 | 0.01187 | 0.01396 | 0.02115 |
| 2 | -0.00055 | 0.00571 | 0.00660 | 0.01170 | 0.01665 | 0.02079 | 0.02464 | 0.02833 | 0.02461 | 0.03324 |
| 3 | 0.00248 | 0.00525 | 0.01117 | 0.01686 | 0.02146 | 0.02562 | 0.02955 | 0.02611 | 0.02751 | 0.03262 |
| 4 | 0.00111 | 0.00890 | 0.01577 | 0.02102 | 0.02559 | 0.02983 | 0.02670 | 0.02599 | 0.03168 | 0.03260 |
| 5 | 0.00358 | 0.01288 | 0.01916 | 0.02431 | 0.02894 | 0.02619 | 0.02523 | 0.02511 | 0.03137 | 0.03163 |
| 6 | 0.00683 | 0.01516 | 0.02120 | 0.02638 | 0.02410 | 0.02278 | 0.02257 | 0.02200 | 0.02799 | 0.03005 |
| 7 | 0.00794 | 0.01569 | 0.02170 | 0.02003 | 0.01817 | 0.01782 | 0.01753 | 0.01797 | 0.02221 | 0.02576 |
| 8 | 0.00975 | 0.01734 | 0.01654 | 0.01377 | 0.01324 | 0.01324 | 0.01153 | 0.01510 | 0.01631 | 0.01904 |
| 9 | 0.01198 | 0.01277 | 0.00819 | 0.00736 | 0.00726 | 0.00553 | 0.00486 | 0.00956 | 0.00830 | 0.01327 |
| 10 | 0.00776 | -0.00238 | -0.00373 | -0.00400 | -0.00576 | -0.00654 | -0.00698 | -0.00204 | -0.00606 | 0.00818 |

TRANSVERSE(V)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.00990 | -0.00165 | -0.00346 | -0.00193 | -0.00797 | -0.01444 | -0.02038 | -0.02624 | -0.02998 | -0.03721 |
| 2 | -0.03844 | -0.03647 | -0.03468 | -0.04112 | -0.04835 | -0.05488 | -0.06121 | -0.06746 | -0.06239 | -0.07989 |
| 3 | -0.03705 | -0.03405 | -0.04107 | -0.04940 | -0.05674 | -0.06370 | -0.07048 | -0.06598 | -0.06220 | -0.08955 |
| 4 | -0.02826 | -0.03662 | -0.04681 | -0.05540 | -0.06325 | -0.07074 | -0.06693 | -0.06631 | -0.05424 | -0.09852 |
| 5 | -0.02383 | -0.03892 | -0.04981 | -0.05906 | -0.06754 | -0.06463 | -0.06377 | -0.06398 | -0.04508 | -0.10480 |
| 6 | -0.01972 | -0.03741 | -0.04929 | -0.05935 | -0.05767 | -0.05646 | -0.05663 | -0.05614 | -0.03320 | -0.10924 |
| 7 | -0.01147 | -0.03144 | -0.04452 | -0.04472 | -0.04303 | -0.04314 | -0.04314 | -0.03854 | -0.01699 | -0.11148 |
| 8 | -0.00613 | -0.02862 | -0.03235 | -0.02989 | -0.02994 | -0.03049 | -0.02825 | -0.02017 | -0.00217 | -0.11186 |
| 9 | -0.00225 | -0.01687 | -0.01301 | -0.01300 | -0.01356 | -0.01139 | -0.01062 | -0.00006 | 0.01456 | -0.11281 |
| 10 | 0.01124 | 0.01859 | 0.01863 | 0.01804 | 0.02012 | 0.02100 | 0.02147 | 0.03342 | 0.04341 | -0.11386 |

VERTICAL(W)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.12128 | 0.12101 | 0.12101 | 0.12101 | 0.12101 | 0.12101 | 0.12101 | 0.12101 | 0.12101 | 0.12101 |
| 2 | 0.05126 | 0.05124 | 0.05124 | 0.05124 | 0.05124 | 0.05124 | 0.05124 | 0.05124 | 0.05124 | 0.05124 |
| 3 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 | 0.02572 |
| 4 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 | 0.00823 |
| 5 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 | -0.00756 |
| 6 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 | -0.02223 |
| 7 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 | -0.03307 |
| 8 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 | -0.04051 |
| 9 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 | -0.04775 |
| 10 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 | -0.05507 |

VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PANEL PRESSURE SINGULARITIES

AXIAL(U)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | 0.00000 | 0.00001 | 0.00003 | 0.00157 | -0.00919 | -0.00091 | -0.00028 |
| 2 | 0.00002 | 0.00056 | 0.00188 | 0.01869 | -0.02887 | -0.00287 | -0.00078 |
| 3 | 0.00191 | 0.00112 | 0.00864 | 0.01959 | -0.02301 | -0.01412 | 0.00458 |
| 4 | 0.00373 | 0.00749 | 0.01312 | 0.02432 | -0.01624 | -0.00639 | -0.00849 |
| 5 | 0.00523 | 0.01032 | 0.01549 | 0.02487 | -0.01378 | -0.01207 | -0.01398 |
| 6 | 0.00696 | 0.01067 | 0.01737 | 0.02352 | -0.01513 | -0.01649 | -0.01586 |
| 7 | 0.00877 | 0.01164 | 0.01605 | 0.02275 | -0.01654 | -0.01731 | -0.01767 |
| 8 | 0.01063 | 0.01159 | 0.01500 | 0.01999 | -0.01666 | -0.01814 | -0.01809 |
| 9 | 0.01071 | 0.01195 | 0.01474 | 0.01773 | -0.01679 | -0.01854 | -0.01922 |
| 10 | 0.01056 | 0.01213 | 0.01421 | 0.01525 | -0.01567 | -0.01862 | -0.01935 |
| 11 | 0.01035 | 0.01055 | 0.00945 | 0.00297 | -0.01311 | -0.01619 | -0.01871 |
| 12 | 0.01334 | 0.01075 | 0.00494 | -0.00197 | -0.00752 | -0.01062 | -0.01215 |
| 13 | 0.00813 | 0.00693 | 0.00446 | 0.00076 | -0.00331 | -0.00647 | -0.00830 |
| 14 | 0.00461 | 0.00386 | 0.00233 | 0.00020 | -0.00187 | -0.00345 | -0.00420 |

TRANSVERSE(V)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | -0.00000 | -0.00001 | -0.00003 | -0.00521 | 0.01925 | 0.00148 | 0.00031 |
| 2 | -0.00009 | 0.00008 | -0.00118 | -0.05481 | 0.07330 | 0.00122 | 0.00056 |
| 3 | 0.00044 | -0.00210 | -0.00471 | -0.10807 | 0.11521 | 0.00488 | 0.00131 |
| 4 | 0.00331 | -0.00296 | -0.01780 | -0.16279 | 0.14329 | 0.00404 | -0.00525 |
| 5 | -0.00174 | -0.00432 | -0.02897 | -0.21270 | 0.17291 | 0.01354 | -0.00131 |
| 6 | -0.00240 | -0.00904 | -0.03908 | -0.25731 | 0.20434 | 0.02431 | 0.00161 |
| 7 | -0.00029 | -0.00843 | -0.04989 | -0.30072 | 0.23366 | 0.03397 | 0.00426 |
| 8 | -0.00153 | -0.01098 | -0.05969 | -0.34145 | 0.26137 | 0.04130 | 0.00616 |
| 9 | -0.00110 | -0.01406 | -0.07121 | -0.37870 | 0.28729 | 0.04935 | 0.00878 |
| 10 | -0.00234 | -0.01697 | -0.08082 | -0.41038 | 0.31072 | 0.05880 | 0.01132 |
| 11 | -0.00348 | -0.02339 | -0.10177 | -0.42632 | 0.34123 | 0.07418 | 0.01615 |
| 12 | -0.00673 | -0.03467 | -0.11093 | -0.42296 | 0.35369 | 0.08204 | 0.01914 |
| 13 | -0.01162 | -0.04556 | -0.12061 | -0.42403 | 0.35963 | 0.08715 | 0.02117 |
| 14 | -0.01477 | -0.05331 | -0.12793 | -0.41451 | 0.33443 | 0.09281 | 0.02409 |

VERTICAL(W)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | 0.00000 | 0.00001 | 0.00006 | -0.00213 | -0.01003 | 0.00113 | 0.00007 |
| 2 | 0.00002 | -0.00006 | 0.00125 | -0.00855 | -0.04184 | 0.00623 | 0.00012 |
| 3 | -0.00219 | 0.00171 | -0.00367 | -0.01227 | -0.06819 | -0.00686 | 0.01779 |
| 4 | -0.00479 | -0.00548 | -0.00755 | -0.01936 | -0.08609 | -0.01059 | 0.01144 |
| 5 | -0.00354 | -0.00732 | -0.01313 | -0.02283 | -0.11022 | -0.02062 | 0.00304 |
| 6 | -0.00345 | -0.00689 | -0.01917 | -0.03152 | -0.12748 | -0.03102 | -0.00627 |
| 7 | -0.00326 | -0.01152 | -0.02348 | -0.03383 | -0.14559 | -0.04199 | -0.01653 |
| 8 | -0.00326 | -0.01297 | -0.02918 | -0.03847 | -0.16383 | -0.05429 | -0.02787 |
| 9 | -0.00385 | -0.01515 | -0.03303 | -0.04296 | -0.18209 | -0.06643 | -0.03913 |
| 10 | -0.00470 | -0.01825 | -0.03925 | -0.04693 | -0.19559 | -0.07714 | -0.05016 |
| 11 | -0.00935 | -0.02579 | -0.04836 | -0.04946 | -0.21321 | -0.09522 | -0.07308 |
| 12 | -0.02873 | -0.04211 | -0.05572 | -0.05151 | -0.21987 | -0.10650 | -0.08339 |
| 13 | -0.04257 | -0.05078 | -0.05825 | -0.05078 | -0.22382 | -0.11472 | -0.09328 |
| 14 | -0.05510 | -0.06010 | -0.06551 | -0.10674 | -0.27143 | -0.12654 | -0.10579 |

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PANEL PRESSURE SINGULARITIES

AXIAL(U)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | 0.00000 | 0.00000 | 0.00000 | 0.02154 | -0.01471 | 0.00000 | 0.00000 |
| 2 | 0.00000 | 0.00000 | 0.01393 | 0.02308 | -0.01590 | -0.00937 | 0.00000 |
| 3 | 0.00454 | 0.01110 | 0.01518 | 0.02052 | -0.01437 | -0.01009 | -0.00946 |
| 4 | 0.01258 | 0.01354 | 0.01606 | 0.01941 | -0.01385 | -0.01222 | -0.01087 |
| 5 | 0.01335 | 0.01452 | 0.01760 | 0.01905 | -0.01381 | -0.01248 | -0.01105 |
| 6 | 0.01364 | 0.01566 | 0.01750 | 0.01843 | -0.01340 | -0.01270 | -0.01150 |
| 7 | 0.01503 | 0.01566 | 0.01727 | 0.01776 | -0.01294 | -0.01259 | -0.01205 |
| 8 | 0.01532 | 0.01609 | 0.01711 | 0.01697 | -0.01241 | -0.01267 | -0.01210 |
| 9 | 0.01532 | 0.01608 | 0.01701 | 0.01554 | -0.01148 | -0.01249 | -0.01198 |
| 10 | 0.01535 | 0.01601 | 0.01597 | 0.01257 | -0.00943 | -0.01188 | -0.01191 |
| 11 | 0.01418 | 0.01368 | 0.00993 | 0.00407 | -0.00350 | -0.00777 | -0.01012 |
| 12 | 0.00871 | 0.00769 | 0.00552 | 0.00211 | -0.00184 | -0.00441 | -0.00560 |
| 13 | 0.00528 | 0.00477 | 0.00340 | 0.00125 | -0.00112 | -0.00275 | -0.00344 |
| 14 | 0.00355 | 0.00308 | 0.00221 | 0.00082 | -0.00069 | -0.00176 | -0.00223 |

TRANSVERSE(V)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | -0.00000 | -0.00000 | -0.00000 | 0.01798 | -0.02638 | -0.00000 | -0.00000 |
| 2 | -0.00000 | -0.00000 | 0.00908 | 0.08085 | -0.08420 | -0.01327 | -0.00000 |
| 3 | 0.00427 | 0.01092 | 0.02426 | 0.13477 | -0.13002 | -0.02688 | -0.00794 |
| 4 | 0.00369 | 0.01478 | 0.03772 | 0.18068 | -0.16866 | -0.03529 | -0.00885 |
| 5 | 0.00533 | 0.01932 | 0.05093 | 0.22301 | -0.20511 | -0.04562 | -0.01128 |
| 6 | 0.00693 | 0.02601 | 0.06438 | 0.26230 | -0.23910 | -0.05609 | -0.01438 |
| 7 | 0.00936 | 0.03194 | 0.07701 | 0.30003 | -0.27176 | -0.06567 | -0.01677 |
| 8 | 0.01064 | 0.03700 | 0.08876 | 0.33579 | -0.30245 | -0.07429 | -0.01921 |
| 9 | 0.01220 | 0.04244 | 0.10065 | 0.36870 | -0.33054 | -0.08305 | -0.02155 |
| 10 | 0.01396 | 0.04794 | 0.11116 | 0.39634 | -0.35328 | -0.09084 | -0.02409 |
| 11 | 0.01618 | 0.05561 | 0.12538 | 0.41924 | -0.36915 | -0.09909 | -0.02697 |
| 12 | 0.01743 | 0.05845 | 0.12986 | 0.42185 | -0.37102 | -0.10161 | -0.02736 |
| 13 | 0.01805 | 0.06060 | 0.13251 | 0.42305 | -0.37197 | -0.10303 | -0.02786 |
| 14 | 0.01852 | 0.06170 | 0.13402 | 0.41394 | -0.34083 | -0.10142 | -0.02778 |

VERTICAL(W)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | -0.00000 | -0.00000 | -0.00000 | -0.04313 | -0.01954 | -0.00000 | -0.00000 |
| 2 | -0.00000 | -0.00000 | -0.02331 | -0.07136 | -0.01458 | -0.01075 | -0.00000 |
| 3 | -0.00562 | -0.01629 | -0.03031 | -0.07704 | 0.00066 | -0.00865 | -0.00699 |
| 4 | -0.01324 | -0.01841 | -0.03371 | -0.07878 | 0.01676 | -0.00428 | -0.00220 |
| 5 | -0.01162 | -0.01748 | -0.03440 | -0.07607 | 0.03537 | 0.00300 | 0.00425 |
| 6 | -0.00905 | -0.01608 | -0.03214 | -0.07253 | 0.05426 | 0.01135 | 0.01091 |
| 7 | -0.00692 | -0.01290 | -0.02928 | -0.06870 | 0.07291 | 0.02016 | 0.01830 |
| 8 | -0.00320 | -0.00961 | -0.02588 | -0.06384 | 0.09159 | 0.02934 | 0.02636 |
| 9 | 0.00093 | -0.00559 | -0.02162 | -0.05656 | 0.11093 | 0.03889 | 0.03463 |
| 10 | 0.00529 | -0.00105 | -0.01589 | -0.04552 | 0.13085 | 0.04902 | 0.04299 |
| 11 | 0.01620 | 0.01148 | 0.00347 | -0.01032 | 0.16733 | 0.07186 | 0.06131 |
| 12 | 0.03157 | 0.02874 | 0.02157 | 0.01030 | 0.18712 | 0.08932 | 0.07833 |
| 13 | 0.04344 | 0.04094 | 0.03474 | 0.02463 | 0.20099 | 0.10194 | 0.09036 |
| 14 | 0.05538 | 0.05435 | 0.05307 | 0.09444 | 0.25970 | 0.11978 | 0.10431 |

VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOURCES

AXIAL(U)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| RCW NO. | | | | | | | |
| 1 | -0.00000 | -0.00000 | -0.00000 | -0.01746 | -0.01483 | -0.00000 | -0.00000 |
| 2 | -0.00000 | -0.00000 | -0.01181 | -0.01010 | -0.00962 | -0.01002 | -0.00000 |
| 3 | -0.00625 | -0.00872 | -0.00864 | -0.00561 | -0.00578 | -0.00773 | -0.01313 |
| 4 | -0.01331 | -0.01218 | -0.00975 | -0.00643 | -0.00803 | -0.01140 | -0.01228 |
| 5 | -0.01091 | -0.01024 | -0.00846 | -0.00500 | -0.00560 | -0.00845 | -0.00946 |
| 6 | -0.00818 | -0.00731 | -0.00524 | -0.00167 | -0.00228 | -0.00529 | -0.00652 |
| 7 | -0.00529 | -0.00424 | -0.00194 | 0.00138 | 0.00079 | -0.00206 | -0.00341 |
| 8 | -0.00221 | -0.00106 | 0.00113 | 0.00388 | 0.00338 | 0.00102 | -0.00019 |
| 9 | 0.00093 | 0.00191 | 0.00385 | 0.00637 | 0.00593 | 0.00380 | 0.00280 |
| 10 | 0.00364 | 0.00460 | 0.00647 | 0.00888 | 0.00845 | 0.00635 | 0.00531 |
| 11 | 0.00839 | 0.00937 | 0.00517 | 0.00283 | 0.00350 | 0.00597 | 0.00996 |
| 12 | 0.00375 | 0.00358 | 0.00314 | 0.00253 | 0.00244 | 0.00273 | 0.00289 |
| 13 | 0.00145 | 0.00132 | 0.C0109 | 0.00088 | 0.00091 | 0.00108 | 0.00119 |
| 14 | 0.00066 | 0.00060 | 0.00050 | 0.00043 | 0.00044 | 0.00051 | 0.00056 |

TRANSVERSE(V)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| RCW NO. | | | | | | | |
| 1 | -0.00000 | -0.00000 | -0.00000 | -0.01254 | -0.01880 | -0.00000 | -0.00000 |
| 2 | -0.00000 | -0.00000 | -0.00686 | -0.02478 | -0.02598 | -0.01151 | -0.00000 |
| 3 | -0.00552 | -0.00751 | -0.01104 | -0.02496 | -0.02455 | -0.01352 | -0.00318 |
| 4 | -0.00003 | -0.00175 | -0.00658 | -0.01576 | -0.01283 | -0.00401 | -0.00102 |
| 5 | -0.00004 | -0.00082 | -0.00351 | -0.00800 | -0.00639 | -0.00269 | -0.00069 |
| 6 | -0.00006 | -0.00063 | -0.00233 | -0.00239 | -0.00071 | -0.00114 | -0.00030 |
| 7 | 0.00004 | -0.00016 | -0.00072 | 0.00340 | 0.00499 | 0.00073 | 0.00020 |
| 8 | 0.00020 | 0.00049 | 0.00107 | 0.00848 | 0.00980 | 0.00254 | 0.00072 |
| 9 | 0.00036 | 0.00104 | 0.00267 | 0.01285 | 0.01394 | 0.00392 | 0.00103 |
| 10 | 0.00042 | 0.00147 | 0.00409 | 0.01681 | 0.01778 | 0.00526 | 0.00139 |
| 11 | 0.00080 | 0.00288 | 0.00416 | 0.00863 | 0.00687 | 0.00276 | 0.00228 |
| 12 | -0.00007 | 0.00004 | 0.00089 | 0.00215 | 0.00175 | 0.00082 | 0.00020 |
| 13 | 0.00015 | 0.00056 | 0.00115 | 0.00163 | 0.00123 | 0.00075 | 0.00023 |
| 14 | 0.00013 | 0.00041 | 0.00072 | 0.00086 | 0.00058 | 0.00044 | 0.00015 |

VERTICAL(W)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| RCW NO. | | | | | | | |
| 1 | -0.00000 | -0.00000 | -0.00000 | 0.03411 | -0.02338 | -0.00000 | -0.00000 |
| 2 | -0.00000 | -0.00000 | 0.02006 | 0.02711 | -0.01888 | -0.01357 | -0.00000 |
| 3 | 0.00799 | 0.01358 | 0.01761 | 0.01797 | -0.01283 | -0.01175 | -0.01298 |
| 4 | 0.01648 | 0.01616 | 0.01563 | 0.01123 | -0.00866 | -0.01219 | -0.01259 |
| 5 | 0.01438 | 0.01391 | 0.01207 | 0.00463 | -0.00395 | -0.00906 | -0.01024 |
| 6 | 0.01150 | 0.01044 | 0.00718 | -0.00210 | 0.00081 | -0.00559 | -0.00754 |
| 7 | 0.00817 | 0.00650 | 0.00204 | -0.00809 | 0.00509 | -0.00193 | -0.00450 |
| 8 | 0.00439 | 0.00223 | -0.00283 | -0.01259 | 0.00836 | 0.00159 | -0.00122 |
| 9 | 0.00038 | -0.00183 | -0.00704 | -0.01662 | 0.01132 | 0.00469 | 0.00190 |
| 10 | -0.00326 | -0.00559 | -0.01098 | -0.02050 | 0.01416 | 0.00757 | 0.00464 |
| 11 | -0.00995 | -0.01261 | -0.00845 | -0.00371 | 0.00291 | 0.00636 | 0.00988 |
| 12 | -0.00519 | -0.00492 | -0.00393 | -0.00118 | 0.00096 | 0.00280 | 0.00329 |
| 13 | -0.00233 | -0.00212 | -0.00154 | -0.00017 | 0.00014 | 0.00117 | 0.00149 |
| 14 | -0.00114 | -0.00100 | -0.00064 | 0.00017 | -0.00015 | 0.00049 | 0.00069 |

PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD = 0.00130 CL = 0.00399 CM = 0.00291

BODY PRESSURE COEFFICIENTS(CP)

| THETA(DEG.) X | 0.0000 | 22.5000 | 45.0000 | 67.5000 | 90.0000 | 112.5000 | 135.0000 | 157.5000 | 180.0000 |
|---|---|---|---|---|---|---|---|---|---|
| 0.0000 | 0.06332 | 0.06803 | 0.08146 | 0.10156 | 0.12527 | 0.14897 | 0.16907 | 0.18250 | 0.18722 |
| 0.7243 | 0.06892 | 0.07363 | 0.08706 | 0.10716 | 0.13087 | 0.15458 | 0.17468 | 0.18810 | 0.19282 |
| 1.4485 | 0.03372 | 0.03851 | 0.05217 | 0.07261 | 0.09672 | 0.12083 | 0.14127 | 0.15492 | 0.15972 |
| 2.1728 | 0.00738 | 0.01100 | 0.02132 | 0.03676 | 0.05497 | 0.07319 | 0.08863 | 0.09894 | 0.10257 |
| 2.8970 | 0.01318 | 0.01663 | 0.02645 | 0.04115 | 0.05849 | 0.07583 | 0.09054 | 0.10036 | 0.10381 |
| 3.6213 | 0.00395 | 0.00705 | 0.01586 | 0.02905 | 0.04461 | 0.06017 | 0.07336 | 0.08217 | 0.08527 |
| 4.3456 | 0.00386 | 0.00683 | 0.01528 | 0.02792 | 0.04284 | 0.05776 | 0.07041 | 0.07886 | 0.08182 |
| 5.0698 | -0.00053 | 0.00223 | 0.01011 | 0.02190 | 0.03580 | 0.04970 | 0.06149 | 0.06937 | 0.07213 |
| 5.7941 | -0.00174 | 0.00091 | 0.00846 | 0.01975 | 0.03307 | 0.04639 | 0.05768 | 0.06523 | 0.06788 |
| 6.5184 | -0.00478 | -0.00228 | 0.00482 | 0.01546 | 0.02800 | 0.04055 | 0.05118 | 0.05829 | 0.06078 |
| 7.2426 | -0.00610 | -0.00372 | 0.00307 | 0.01322 | 0.02520 | 0.03717 | 0.04733 | 0.05411 | 0.05649 |
| 7.9669 | -0.00869 | -0.00644 | -0.00004 | 0.00953 | 0.02083 | 0.03213 | 0.04170 | 0.04810 | 0.05035 |
| 8.6911 | -0.00988 | -0.00775 | -0.00168 | 0.00741 | 0.01812 | 0.02884 | 0.03792 | 0.04399 | 0.04612 |
| 9.4154 | -0.01211 | -0.01011 | -0.00440 | 0.00415 | 0.01423 | 0.02431 | 0.03286 | 0.03857 | 0.04057 |
| 10.1397 | -0.01341 | -0.01153 | -0.00616 | 0.00188 | 0.01136 | 0.02084 | 0.02888 | 0.03425 | 0.03613 |
| 10.8639 | -0.01601 | -0.01425 | -0.00924 | -0.00175 | 0.00709 | 0.01594 | 0.02343 | 0.02844 | 0.03020 |
| 11.5882 | -0.01743 | -0.01581 | -0.01118 | -0.00426 | 0.00391 | 0.01208 | 0.01900 | 0.02363 | 0.02525 |
| 12.3124 | -0.02077 | -0.01928 | -0.01505 | -0.00872 | -0.00125 | 0.00622 | 0.01255 | 0.01678 | 0.01827 |
| 13.0367 | -0.02210 | -0.02078 | -0.01701 | -0.01138 | -0.00473 | 0.00192 | 0.00756 | 0.01132 | 0.01265 |
| 13.7610 | -0.02476 | -0.02360 | -0.02028 | -0.01530 | -0.00944 | -0.00358 | 0.00139 | 0.00471 | 0.00588 |
| 14.5023 | -0.02727 | -0.02630 | -0.02353 | -0.01939 | -0.01451 | -0.00963 | -0.00549 | -0.00273 | -0.00175 |
| 15.2436 | -0.03820 | -0.03744 | -0.03531 | -0.03211 | -0.02833 | -0.02456 | -0.02136 | -0.01922 | -0.01847 |
| 15.9849 | -0.03942 | -0.03902 | -0.03787 | -0.03616 | -0.03414 | -0.03212 | -0.03041 | -0.02926 | -0.02886 |
| 16.7262 | -0.02689 | -0.02674 | -0.02631 | -0.02565 | -0.02488 | -0.02412 | -0.02346 | -0.02303 | -0.02288 |
| 17.4675 | -0.02161 | -0.02163 | -0.02167 | -0.02174 | -0.02182 | -0.02189 | -0.02196 | -0.02200 | -0.02202 |
| 18.2088 | -0.01660 | -0.01672 | -0.01705 | -0.01756 | -0.01815 | -0.01874 | -0.01925 | -0.01958 | -0.01970 |
| 18.9501 | -0.01334 | -0.01351 | -0.01400 | -0.01474 | -0.01561 | -0.01648 | -0.01721 | -0.01771 | -0.01788 |
| 19.6914 | -0.01089 | -0.01108 | -0.01163 | -0.01246 | -0.01343 | -0.01441 | -0.01523 | -0.01578 | -0.01598 |
| 20.4327 | -0.00917 | -0.00936 | -0.00991 | -0.01073 | -0.01169 | -0.01266 | -0.01348 | -0.01402 | -0.01422 |
| 21.1740 | -0.00792 | -0.00810 | -0.00860 | -0.00936 | -0.01025 | -0.01113 | -0.01189 | -0.01239 | -0.01257 |
| 21.9153 | -0.00701 | -0.00716 | -0.00760 | -0.00826 | -0.00904 | -0.00982 | -0.01048 | -0.01092 | -0.01108 |
| 22.6566 | -0.00632 | -0.00645 | -0.00682 | -0.00738 | -0.00803 | -0.00869 | -0.00924 | -0.00962 | -0.00975 |
| 23.3979 | -0.00578 | -0.00588 | -0.00619 | -0.00664 | -0.00718 | -0.00771 | -0.00816 | -0.00847 | -0.00857 |
| 24.1392 | -0.00533 | -0.00542 | -0.00566 | -0.00602 | -0.00645 | -0.00687 | -0.00723 | -0.00747 | -0.00756 |
| 24.8805 | -0.00495 | -0.00502 | -0.00521 | -0.00549 | -0.00582 | -0.00615 | -0.00643 | -0.00662 | -0.00668 |
| 25.6218 | -0.00462 | -0.00467 | -0.00481 | -0.00502 | -0.00528 | -0.00553 | -0.00574 | -0.00589 | -0.00594 |
| 26.3631 | -0.00431 | -0.00435 | -0.00446 | -0.00462 | -0.00481 | -0.00500 | -0.00516 | -0.00527 | -0.00531 |
| 27.1044 | -0.00402 | -0.00404 | -0.00412 | -0.00424 | -0.00439 | -0.00453 | -0.00465 | -0.00473 | -0.00476 |
| 27.8457 | -0.00379 | -0.00381 | -0.00387 | -0.00396 | -0.00407 | -0.00417 | -0.00426 | -0.00432 | -0.00434 |
| 28.5870 | -0.00339 | -0.00340 | -0.00345 | -0.00352 | -0.00360 | -0.00368 | -0.00375 | -0.00379 | -0.00381 |
| 29.3283 | -0.00371 | -0.00372 | -0.00376 | -0.00381 | -0.00387 | -0.00393 | -0.00399 | -0.00402 | -0.00403 |
| 30.0696 | -0.00881 | -0.00885 | -0.00895 | -0.00910 | -0.00928 | -0.00946 | -0.00961 | -0.00971 | -0.00974 |
| 30.8109 | -0.01632 | -0.01646 | -0.01687 | -0.01747 | -0.01817 | -0.01888 | -0.01948 | -0.01988 | -0.02002 |
| 31.5522 | -0.02026 | -0.02059 | -0.02151 | -0.02289 | -0.02451 | -0.02614 | -0.02752 | -0.02844 | -0.02876 |
| 32.2935 | -0.02060 | -0.02115 | -0.02271 | -0.02505 | -0.02781 | -0.03057 | -0.03291 | -0.03447 | -0.03502 |
| 33.0348 | -0.02260 | -0.02341 | -0.02569 | -0.02912 | -0.03315 | -0.03719 | -0.04062 | -0.04290 | -0.04371 |
| 33.7761 | -0.02435 | -0.02548 | -0.02867 | -0.03346 | -0.03910 | -0.04474 | -0.04953 | -0.05272 | -0.05385 |
| 34.5174 | -0.01577 | -0.01721 | -0.02133 | -0.02749 | -0.03476 | -0.04203 | -0.04819 | -0.05230 | -0.05375 |
| 35.2587 | 0.00286 | 0.00124 | -0.00334 | -0.01020 | -0.01830 | -0.02639 | -0.03326 | -0.03784 | -0.03945 |

INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD = 0.00177 CL = 0.01667 CM = -0.36813

BODY PANEL PRESSURE COEFFICIENT(CP)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| ROW NO. | | | | | | | |
| 1 | -0.00000 | -0.00002 | -0.00006 | -0.01130 | 0.07745 | 0.00183 | 0.00055 |
| 2 | -0.00004 | -0.00112 | -0.00800 | -0.06334 | 0.10880 | 0.04453 | 0.00156 |
| 3 | -0.00039 | -0.00699 | -0.03037 | -0.06901 | 0.08632 | 0.06386 | 0.03601 |
| 4 | -0.00599 | -0.01771 | -0.03885 | -0.07459 | 0.07625 | 0.06003 | 0.06327 |
| 5 | -0.01533 | -0.02922 | -0.04925 | -0.07783 | 0.06639 | 0.06600 | 0.06898 |
| 6 | -0.02483 | -0.03804 | -0.05924 | -0.08058 | 0.06163 | 0.06895 | 0.06776 |
| 7 | -0.03703 | -0.04612 | -0.06275 | -0.08378 | 0.05739 | 0.06391 | 0.06625 |
| 8 | -0.04748 | -0.05323 | -0.06649 | -0.08168 | 0.05137 | 0.05957 | 0.06076 |
| 9 | -0.05391 | -0.05987 | -0.07120 | -0.07928 | 0.04468 | 0.05445 | 0.05680 |
| 10 | -0.05911 | -0.06550 | -0.07329 | -0.07342 | 0.03328 | 0.04829 | 0.05191 |
| 11 | -0.06585 | -0.06719 | -0.04911 | -0.01974 | 0.02623 | 0.03596 | 0.03773 |
| 12 | -0.05160 | -0.04405 | -0.02720 | -0.00533 | 0.01384 | 0.02459 | 0.02972 |
| 13 | -0.02972 | -0.02603 | -0.01790 | -0.00577 | 0.00704 | 0.01627 | 0.02109 |
| 14 | -0.01763 | -0.01510 | -0.01010 | -0.00291 | 0.00424 | 0.00940 | 0.01174 |

BODY PANEL SLOPE(DR/DX)

| THETA(DEG.) | 12.5000 | 37.5000 | 62.5000 | 88.7967 | 116.0950 | 142.5000 | 167.5000 |
|---|---|---|---|---|---|---|---|
| RCW NO. | | | | | | | |
| 1 | -0.08032 | -0.06116 | -0.02643 | 0.01972 | 0.06798 | 0.10500 | 0.12416 |
| 2 | -0.09482 | -0.07566 | -0.04094 | 0.00491 | 0.05348 | 0.09050 | 0.10965 |
| 3 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 4 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 5 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 6 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 7 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 8 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 9 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 10 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 11 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 12 | -0.10224 | -0.08308 | -0.04835 | -0.00257 | 0.04606 | 0.08308 | 0.10224 |
| 13 | -0.10237 | -0.08321 | -0.04849 | -0.00270 | 0.04593 | 0.08295 | 0.10211 |
| 14 | -0.12598 | -0.10682 | -0.07209 | -0.02631 | 0.02232 | 0.05934 | 0.07850 |

PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN PRESENCE OF BODY

CD = 0.01949 CL = 0.18361 CM = -5.42608

SPANWISE CD DISTRIBUTION(B/2Q * DD/DY)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.00186 | 0.00192 | 0.00201 | 0.00208 | 0.00212 | 0.00215 | 0.00222 | 0.00242 | 0.00260 | 0.00213 |

SPANWISE CL DISTRIBUTION(B/2Q * DL/DY)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 0.01680 | 0.01779 | 0.01867 | 0.01941 | 0.02002 | 0.02059 | 0.02141 | 0.02305 | 0.02455 | 0.02015 |

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.16579 | -0.21452 | -0.24268 | -0.25782 | -0.28390 | -0.30588 | -0.32415 | -0.34268 | -0.35644 | -0.56583 |
| 2 | -0.13484 | -0.15824 | -0.15489 | -0.18337 | -0.20003 | -0.21325 | -0.22674 | -0.23586 | -0.24208 | -0.35603 |
| 3 | -0.12453 | -0.13811 | -0.17466 | -0.15877 | -0.17035 | -0.18187 | -0.18886 | -0.18594 | -0.21296 | -0.27307 |
| 4 | -0.11876 | -0.12168 | -0.13465 | -0.14681 | -0.15633 | -0.16163 | -0.15786 | -0.15751 | -0.20251 | -0.20705 |
| 5 | -0.10660 | -0.11828 | -0.13002 | -0.13998 | -0.14111 | -0.13828 | -0.13257 | -0.14280 | -0.18144 | -0.17239 |
| 6 | -0.10478 | -0.11510 | -0.12141 | -0.13059 | -0.12053 | -0.11014 | -0.11602 | -0.12862 | -0.14999 | -0.14371 |
| 7 | -0.10168 | -0.10454 | -0.11355 | -0.10181 | -0.09604 | -0.09265 | -0.09650 | -0.11126 | -0.11757 | -0.11524 |
| 8 | -0.09239 | -0.09964 | -0.08509 | -0.07717 | -0.07169 | -0.07441 | -0.07352 | -0.09540 | -0.08590 | -0.08092 |
| 9 | -0.08799 | -0.06942 | -0.05694 | -0.04836 | -0.05112 | -0.04631 | -0.05180 | -0.06944 | -0.05392 | -0.04756 |
| 10 | -0.05000 | -0.02157 | -0.01570 | -0.01678 | -0.00975 | -0.01161 | -0.01471 | -0.02991 | -0.01117 | -0.01279 |

LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.18348 | 0.20371 | 0.22155 | 0.24793 | 0.26083 | 0.27240 | 0.28208 | 0.28676 | 0.29986 | 0.47593 |
| 2 | 0.07736 | 0.08616 | 0.11668 | 0.11394 | 0.11750 | 0.12032 | 0.11950 | 0.12080 | 0.14217 | 0.22797 |
| 3 | 0.04907 | 0.05701 | 0.04242 | 0.07438 | 0.07425 | 0.07106 | 0.06950 | 0.07984 | 0.10821 | 0.14360 |
| 4 | 0.03860 | 0.05056 | 0.05332 | 0.05040 | 0.04670 | 0.04373 | 0.04940 | 0.05892 | 0.07631 | 0.07650 |
| 5 | 0.03956 | 0.04148 | 0.03677 | 0.03087 | 0.02991 | 0.03162 | 0.03834 | 0.04325 | 0.05240 | 0.04208 |
| 6 | 0.03831 | 0.03155 | 0.02705 | 0.01607 | 0.02224 | 0.03060 | 0.02689 | 0.03628 | 0.03620 | 0.02124 |
| 7 | 0.03421 | 0.02881 | 0.01538 | 0.02014 | 0.02143 | 0.02299 | 0.02328 | 0.03711 | 0.02290 | 0.00730 |
| 8 | 0.03278 | 0.01598 | 0.01929 | 0.01984 | 0.02119 | 0.01774 | 0.02499 | 0.02913 | 0.01181 | -0.00084 |
| 9 | 0.01947 | 0.01885 | 0.01997 | 0.02243 | 0.01737 | 0.02173 | 0.02534 | 0.02280 | 0.00449 | -0.01095 |
| 10 | 0.02125 | 0.03102 | 0.02910 | 0.02558 | 0.03058 | 0.02944 | 0.03480 | 0.02213 | 0.01354 | -0.02950 |

WING PANEL PRESSURE DIFFERENCE(CL)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | 0.34927 | 0.41823 | 0.46423 | 0.50575 | 0.54473 | 0.57828 | 0.60622 | 0.62944 | 0.65630 | 1.04576 |
| 2 | 0.21220 | 0.24440 | 0.27157 | 0.29731 | 0.31753 | 0.33357 | 0.34624 | 0.35667 | 0.38426 | 0.58400 |
| 3 | 0.17360 | 0.19512 | 0.21708 | 0.23315 | 0.24459 | 0.25293 | 0.25835 | 0.26579 | 0.32118 | 0.41667 |
| 4 | 0.15736 | 0.17224 | 0.18797 | 0.19721 | 0.20303 | 0.20536 | 0.20726 | 0.21642 | 0.27881 | 0.28354 |
| 5 | 0.14616 | 0.15976 | 0.16679 | 0.17084 | 0.17102 | 0.16990 | 0.17090 | 0.18605 | 0.23384 | 0.21447 |
| 6 | 0.14309 | 0.14666 | 0.14846 | 0.14666 | 0.14277 | 0.14074 | 0.14292 | 0.16490 | 0.18619 | 0.16495 |
| 7 | 0.13589 | 0.13335 | 0.12892 | 0.12195 | 0.11746 | 0.11564 | 0.11978 | 0.14837 | 0.14047 | 0.12253 |
| 8 | 0.12517 | 0.11562 | 0.10438 | 0.09701 | 0.09287 | 0.09216 | 0.09851 | 0.12452 | 0.09771 | 0.08009 |
| 9 | 0.10746 | 0.08827 | 0.07691 | 0.07079 | 0.06849 | 0.06804 | 0.07714 | 0.09224 | 0.05842 | 0.03661 |
| 10 | 0.07125 | 0.05259 | 0.04480 | 0.04235 | 0.04033 | 0.04105 | 0.04951 | 0.05205 | 0.02472 | -0.01672 |

UPPER SURFACE WING PANEL SLOPE(DZ/DX)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 | -0.03734 |
| 2 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06962 | -0.06562 |
| 3 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 | -0.08838 |
| 4 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 | -0.10461 |
| 5 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 | -0.11995 |
| 6 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 | -0.13395 |
| 7 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 | -0.14162 |
| 8 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 | -0.14885 |
| 9 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 | -0.15609 |
| 10 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 | -0.16348 |

LOWER SURFACE WING PANEL SLOPE(DZ/DX)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 | -0.17210 |
| 2 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 | -0.13982 |
| 3 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 | -0.12106 |
| 4 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 | -0.10483 |
| 5 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 | -0.08949 |
| 6 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 | -0.07548 |
| 7 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 | -0.06782 |
| 8 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 | -0.06059 |
| 9 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 | -0.05335 |
| 10 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 | -0.04596 |

WING CAMBER SLOPE(DZ/DX)

| SPANWISE STATION | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CHORDWISE STATION | | | | | | | | | | |
| 1 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 2 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 3 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 4 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 5 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 6 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 7 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 8 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 9 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 |
| 10 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.10472 | -0.1C472 |

WING CHORD LENGTHS(C)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 9.21046 | 9.21070 | 9.21094 | 9.21118 | 9.21142 | 9.21166 | 9.21190 | 9.21215 | 9.21239 | 6.14167 |

FORCES AND MOMENTS ON WING-BODY COMBINATION

CD = 0.02256 CL = 0.20427 CM = -5.79131

POINTS

| X | Y | Z | UP | VP | WP | U | V | W | CP |
|---|---|---|---|---|---|---|---|---|---|
| 24.00000 | 4.00000 | -1.00000 | -0.04106 | 0.10128 | -0.05855 | 0.95346 | 0.10128 | 0.04597 | 0.08211 |
| 24.00000 | 4.00000 | -0.50000 | -0.04567 | 0.12937 | -0.08649 | 0.94886 | 0.12937 | 0.01803 | 0.09133 |
| 24.00000 | 4.35350 | -0.64650 | -0.06030 | 0.15698 | -0.04298 | 0.93422 | 0.15698 | 0.06155 | 0.12061 |
| 24.00000 | 4.50000 | -1.00000 | -0.04645 | 0.11726 | -0.01784 | 0.94807 | 0.11726 | 0.08669 | 0.09290 |
| 24.00000 | 4.35350 | -1.35350 | -0.03830 | 0.08974 | -0.03034 | 0.95622 | 0.08974 | 0.07418 | 0.07461 |
| 24.00000 | 4.00000 | -1.50000 | -0.03553 | 0.07721 | -0.04744 | 0.95899 | 0.07721 | 0.05709 | 0.07107 |
| 24.00000 | 3.64650 | -1.35350 | -0.03304 | 0.07027 | -0.06294 | 0.96148 | 0.07027 | 0.04159 | 0.06609 |
| 24.00000 | 3.50000 | -1.00000 | -0.03229 | 0.07365 | -0.07657 | 0.96224 | 0.07365 | 0.02796 | 0.06457 |
| 24.00000 | 3.64650 | -0.64650 | -0.03502 | 0.08909 | -0.08500 | 0.95950 | 0.08909 | 0.01953 | 0.07004 |

GRIDS

COMPUTE** 2**GRIDS

OPTION REQUESTED---RECTANGULAR GRID

X-Z CUT, FOR THIS CUT Y= 1.20000

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 30.000000 | 1.200000 | -1.200000 | 0.989934E 00 | -0.570023E-01 | 0.104333E 00 | 0.917638E-02 |
| 30.500000 | 1.200000 | -1.200000 | 0.992241E 00 | -0.600857E-01 | 0.107695E 00 | 0.456158E-02 |
| 31.000000 | 1.200000 | -1.200000 | 0.995679E 00 | -0.635194E-01 | 0.112374E 00 | -0.231379E-02 |
| 31.500000 | 1.200000 | -1.200000 | 0.998674E 00 | -0.669344E-01 | 0.117447E 00 | -0.830407E-02 |
| 32.000000 | 1.200000 | -1.200000 | 0.100098E 01 | -0.708257E-01 | 0.122179E 00 | -0.129208E-01 |
| 32.500000 | 1.200000 | -1.200000 | 0.100344E 01 | -0.752087E-01 | 0.126561E 00 | -0.178279E-01 |
| 33.000000 | 1.200000 | -1.200000 | 0.100548E 01 | -0.783316E-01 | 0.130438E 00 | -0.219261E-01 |
| 33.500000 | 1.200000 | -1.200000 | 0.100700E 01 | -0.815329E-01 | 0.134203E 00 | -0.249633E-01 |
| 34.000000 | 1.200000 | -1.200000 | 0.100919E 01 | -0.860532E-01 | 0.138596E 00 | -0.293384E-01 |
| 34.500000 | 1.200000 | -1.200000 | 0.101167E 01 | -0.913022E-01 | 0.143104E 00 | -0.342880E-01 |
| 30.000000 | 1.200000 | -0.700000 | 0.992804E 00 | -0.188484E 00 | 0.132079E 00 | 0.343511E-02 |
| 30.500000 | 1.200000 | -0.700000 | 0.996728E 00 | -0.193886E 00 | 0.136188E 00 | -0.441172E-02 |
| 31.000000 | 1.200000 | -0.700000 | 0.100057E 01 | -0.198870E 00 | 0.140716E 00 | -0.120993E-01 |
| 31.500000 | 1.200000 | -0.700000 | 0.100313E 01 | -0.203528E 00 | 0.144857E 00 | -0.172204E-01 |
| 32.000000 | 1.200000 | -0.700000 | 0.100498E 01 | -0.208356E 00 | 0.148345E 00 | -0.209249E-01 |
| 32.500000 | 1.200000 | -0.700000 | 0.100701E 01 | -0.213255E 00 | 0.151364E 00 | -0.249766E-01 |
| 33.000000 | 1.200000 | -0.700000 | 0.100988E 01 | -0.217647E 00 | 0.154221E 00 | -0.287181E-01 |
| 33.500000 | 1.200000 | -0.700000 | 0.101073E 01 | -0.223207E 00 | 0.157269E 00 | -0.324099E-01 |
| 34.000000 | 1.200000 | -0.700000 | 0.101305E 01 | -0.229433E 00 | 0.160243E 00 | -0.370472E-01 |
| 34.500000 | 1.200000 | -0.700000 | 0.101447E 01 | -0.235280E 00 | 0.162273E 00 | -0.398961E-01 |

X-Z CUT, FOR THIS CUT Y= 2.20000

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 30.000000 | 2.200000 | -1.200000 | 0.990985E 00 | -0.341504E-01 | 0.409533E-01 | 0.707457E-02 |
| 30.500000 | 2.200000 | -1.200000 | 0.990063E 00 | -0.337229E-01 | 0.427549E-01 | 0.891712E-02 |
| 31.000000 | 2.200000 | -1.200000 | 0.989512E 00 | -0.339817E-01 | 0.445742E-01 | 0.100207E-01 |
| 31.500000 | 2.200000 | -1.200000 | 0.990502E 00 | -0.346020E-01 | 0.470522E-01 | 0.804067E-02 |
| 32.000000 | 2.200000 | -1.200000 | 0.993469E 00 | -0.384465E-01 | 0.503422E-01 | 0.210673E-02 |
| 32.500000 | 2.200000 | -1.200000 | 0.996106E 00 | -0.430525E-01 | 0.535883E-01 | -0.314909E-02 |
| 33.000000 | 2.200000 | -1.200000 | 0.998468E 00 | -0.467513E-01 | 0.569910E-01 | -0.789310E-02 |
| 33.500000 | 2.200000 | -1.200000 | 0.100020E 01 | -0.497416E-01 | 0.598296E-01 | -0.113549E-01 |
| 34.000000 | 2.200000 | -1.200000 | 0.100210E 01 | -0.531640E-01 | 0.624276E-01 | -0.151644E-01 |
| 34.500000 | 2.200000 | -1.200000 | 0.100334E 01 | -0.562142E-01 | 0.646170E-01 | -0.176443E-01 |
| 30.000000 | 2.200000 | -0.700000 | 0.993251E 00 | -0.364391E-01 | 0.172873E-01 | 0.254232E-02 |
| 30.500000 | 2.200000 | -0.700000 | 0.992023E 00 | -0.356816E-01 | 0.195971E-01 | 0.499710E-02 |
| 31.000000 | 2.200000 | -0.700000 | 0.991838E 00 | -0.355215E-01 | 0.219863E-01 | 0.536682E-02 |
| 31.500000 | 2.200000 | -0.700000 | 0.993379E 00 | -0.370613E-01 | 0.247158E-01 | 0.228539E-02 |
| 32.000000 | 2.200000 | -0.700000 | 0.996538E 00 | -0.423572E-01 | 0.276585E-01 | -0.403212E-02 |
| 32.500000 | 2.200000 | -0.700000 | 0.999427E 00 | -0.469981E-01 | 0.306784E-01 | -0.981105E-02 |
| 33.000000 | 2.200000 | -0.700000 | 0.100107E 01 | -0.502949E-01 | 0.331996E-01 | -0.131000E-01 |
| 33.500000 | 2.200000 | -0.700000 | 0.100261E 01 | -0.532604E-01 | 0.353834E-01 | -0.161818E-01 |
| 34.000000 | 2.200000 | -0.700000 | 0.100414E 01 | -0.565240E-01 | 0.372569E-01 | -0.192378E-01 |
| 34.500000 | 2.200000 | -0.700000 | 0.100539E 01 | -0.599930E-01 | 0.390190E-01 | -0.217425E-01 |

X-Z CUT, FOR THIS CUT Y= 3.20000

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 30.000000 | 3.200000 | -1.200000 | 0.988189E 00 | -0.803656E-02 | 0.288318E-01 | 0.126649E-01 |
| 30.500000 | 3.200000 | -1.200000 | 0.989045E 00 | -0.877997E-02 | 0.323951E-01 | 0.109544E-01 |
| 31.000000 | 3.200000 | -1.200000 | 0.989705E 00 | -0.106019E-01 | 0.350450E-01 | 0.963442E-02 |
| 31.500000 | 3.200000 | -1.200000 | 0.989551E 00 | -0.122944E-01 | 0.369454E-01 | 0.994232E-02 |
| 32.000000 | 3.200000 | -1.200000 | 0.989243E 00 | -0.122439E-01 | 0.392729E-01 | 0.105575E-01 |
| 32.500000 | 3.200000 | -1.200000 | 0.989317E 00 | -0.126821E-01 | 0.415194E-01 | 0.104095E-01 |
| 33.000000 | 3.200000 | -1.200000 | 0.990214E 00 | -0.145600E-01 | 0.438990E-01 | 0.861502E-02 |
| 33.500000 | 3.200000 | -1.200000 | 0.992827E 00 | -0.192454E-01 | 0.467426E-01 | 0.338951E-02 |
| 34.000000 | 3.200000 | -1.200000 | 0.995451E 00 | -0.235755E-01 | 0.497268E-01 | -0.185870E-02 |
| 34.500000 | 3.200000 | -1.200000 | 0.997736E 00 | -0.263678E-01 | 0.527333E-01 | -0.642898E-02 |
| 30.000000 | 3.200000 | -0.700000 | 0.992424E 00 | -0.119636E-01 | 0.185772E-01 | 0.419538E-02 |
| 30.500000 | 3.200000 | -0.700000 | 0.992248E 00 | -0.129206E-01 | 0.225627E-01 | 0.454846E-02 |
| 31.000000 | 3.200000 | -0.700000 | 0.992456E 00 | -0.146440E-01 | 0.261315E-01 | 0.413249E-02 |
| 31.500000 | 3.200000 | -0.700000 | 0.992345E 00 | -0.151453E-01 | 0.294851E-01 | 0.435471E-02 |
| 32.000000 | 3.200000 | -0.700000 | 0.991749E 00 | -0.144022E-01 | 0.323928E-01 | 0.554651E-02 |
| 32.500000 | 3.200000 | -0.700000 | 0.991759E 00 | -0.145617E-01 | 0.351001E-01 | 0.552632E-02 |
| 33.000000 | 3.200000 | -0.700000 | 0.992862E 00 | -0.169365E-01 | 0.377980E-01 | 0.331899E-02 |
| 33.500000 | 3.200000 | -0.700000 | 0.995720E 00 | -0.219002E-01 | 0.406324E-01 | -0.239646E-02 |
| 34.000000 | 3.200000 | -0.700000 | 0.998451E 00 | -0.256426E-01 | 0.434677E-01 | -0.785911E-02 |
| 34.500000 | 3.200000 | -0.700000 | 0.999798E 00 | -0.281874E-01 | 0.457221E-01 | -0.105531E-01 |

OPTION REQUESTED---RECTANGULAR GRID

X-Z CUT, FOR THIS CUT Y= 0.00000

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 1490.000000 | 0.000000 | -1000.000000 | 0.994572E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1491.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1492.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1493.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1494.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1495.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1496.000000 | 0.000000 | -1000.000000 | 0.994522E 00 | 0.000000E-38 | 0.104528E 00 | -0.000000E-38 |
| 1497.000000 | 0.000000 | -1000.000000 | 0.993653E 00 | 0.258450E-11 | 0.103223E 00 | 0.173750E-02 |
| 1498.000000 | 0.000000 | -1000.000000 | 0.993357E 00 | 0.345393E-11 | 0.102767E 00 | 0.232914E-02 |
| 1499.000000 | 0.000000 | -1000.000000 | 0.993433E 00 | 0.310122E-11 | 0.102949E 00 | 0.217721E-02 |
| 1500.000000 | 0.000000 | -1000.000000 | 0.993632E 00 | 0.268112E-11 | 0.103177E 00 | 0.177891E-02 |
| 1501.000000 | 0.000000 | -1000.000000 | 0.993701E 00 | 0.248107E-11 | 0.103278E 00 | 0.164109E-02 |
| 1502.000000 | 0.000000 | -1000.000000 | 0.993745E 00 | 0.230182E-11 | 0.103365E 00 | 0.155405E-02 |
| 1503.000000 | 0.000000 | -1000.000000 | 0.993815E 00 | 0.209944E-11 | 0.103468E 00 | 0.141359E-02 |
| 1504.000000 | 0.000000 | -1000.000000 | 0.993892E 00 | 0.179844E-11 | 0.103616E 00 | 0.125941E-02 |
| 1505.000000 | 0.000000 | -1000.000000 | 0.994019E 00 | 0.150747E-11 | 0.103766E 00 | 0.100506E-02 |
| 1506.000000 | 0.000000 | -1000.000000 | 0.994136E 00 | 0.116058E-11 | 0.103943E 00 | 0.771667E-03 |
| 1507.000000 | 0.000000 | -1000.000000 | 0.994262E 00 | 0.788293E-12 | 0.104133E 00 | 0.518991E-03 |
| 1508.000000 | 0.000000 | -1000.000000 | 0.994421E 00 | 0.331798E-12 | 0.104362E 00 | 0.201684E-03 |
| 1509.000000 | 0.000000 | -1000.000000 | 0.994594E 00 | -0.182483E-12 | 0.104622E 00 | -0.144498E-03 |
| 1510.000000 | 0.000000 | -1000.000000 | 0.994880E 00 | -0.111386E-11 | 0.105092E 00 | -0.716940E-03 |
| 1511.000000 | 0.000000 | -1000.000000 | 0.993486E 00 | -0.183754E-11 | 0.102998E 00 | 0.207200E-02 |
| 1512.000000 | 0.000000 | -1000.000000 | 0.993035E 00 | -0.145243E-11 | 0.101827E 00 | 0.297438E-02 |
| 1513.000000 | 0.000000 | -1000.000000 | 0.993066E 00 | -0.134618E-11 | 0.102750E 00 | 0.291263E-02 |
| 1514.000000 | 0.000000 | -1000.000000 | 0.994060E 00 | -0.116481E-11 | 0.103915E 00 | 0.923599E-03 |
| 1515.000000 | 0.000000 | -1000.000000 | 0.993543E 00 | -0.979960E-12 | 0.103225E 00 | 0.195847E-02 |
| 1516.000000 | 0.000000 | -1000.000000 | 0.993713E 00 | -0.834508E-12 | 0.103401E 00 | 0.161771E-02 |
| 1517.000000 | 0.000000 | -1000.000000 | 0.993533E 00 | -0.705972E-12 | 0.103072E 00 | 0.197748E-02 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1518.000000 | 0.000000 | -1000.000000 | 0.993257E 00 | -0.607332E-12 | 0.102529E 00 | 0.253037E-02 |
| 1519.000000 | 0.000000 | -1000.000000 | 0.993488E 00 | -0.512281E-12 | 0.102957E 00 | 0.206863E-02 |
| 1520.000000 | 0.000000 | -1000.000000 | 0.993474E 00 | -0.453906E-12 | 0.102892E 00 | 0.209487E-02 |
| 1521.000000 | 0.000000 | -1000.000000 | 0.993223E 00 | -0.403420E-12 | 0.102669E 00 | 0.259770E-02 |
| 1522.000000 | 0.000000 | -1000.000000 | 0.993347E 00 | -0.356067E-12 | 0.102802E 00 | 0.234910E-02 |
| 1523.000000 | 0.000000 | -1000.000000 | 0.993378E 00 | -0.325649E-12 | 0.103012E 00 | 0.228770E-02 |
| 1524.000000 | 0.000000 | -1000.000000 | 0.993589E 00 | -0.302837E-12 | 0.102999E 00 | 0.186597E-02 |
| 1525.000000 | 0.000000 | -1000.000000 | 0.993732E 00 | -0.635104E-12 | 0.103311E 00 | 0.157908E-02 |
| 1526.000000 | 0.000000 | -1000.000000 | 0.994017E 00 | -0.133060E-11 | 0.103734E 00 | 0.100962E-02 |
| 1527.000000 | 0.000000 | -1000.000000 | 0.994185E 00 | -0.182383E-11 | 0.103964E 00 | 0.673938E-03 |
| 1528.000000 | 0.000000 | -1000.000000 | 0.994433E 00 | -0.233242E-11 | 0.104487E 00 | 0.177965E-03 |
| 1529.000000 | 0.000000 | -1000.000000 | 0.994606E 00 | -0.283740E-11 | 0.104612E 00 | -0.167787E-03 |
| 1530.000000 | 0.000000 | -1000.000000 | 0.994288E 00 | -0.241122E-11 | 0.104301E 00 | 0.467738E-03 |
| 1531.000000 | 0.000000 | -1000.000000 | 0.992059E 00 | 0.190878E-11 | 0.100855E 00 | 0.492662E-02 |
| 1532.000000 | 0.000000 | -1000.000000 | 0.992202E 00 | 0.713351E-12 | 0.100985E 00 | 0.463922E-02 |
| 1533.000000 | 0.000000 | -1000.000000 | 0.992703E 00 | 0.367806E-12 | 0.101772E 00 | 0.363848E-02 |
| 1534.000000 | 0.000000 | -1000.000000 | 0.993580E 00 | 0.224168E-12 | 0.103128E 00 | 0.188456E-02 |
| 1535.000000 | 0.000000 | -1000.000000 | 0.994623E 00 | 0.143851E-12 | 0.104616E 00 | -0.203189E-03 |
| 1536.000000 | 0.000000 | -1000.000000 | 0.995711E 00 | 0.845004E-13 | 0.106345E 00 | -0.237769E-02 |
| 1537.000000 | 0.000000 | -1000.000000 | 0.997737E 00 | 0.594470E-13 | 0.109395E 00 | -0.643097E-02 |
| 1538.000000 | 0.000000 | -1000.000000 | 0.997040E 00 | 0.292057E-13 | 0.108306E 00 | -0.503599E-02 |
| 1539.000000 | 0.000000 | -1000.000000 | 0.996286E 00 | 0.123629E-13 | 0.107168E 00 | -0.352768E-02 |
| 1540.000000 | 0.000000 | -1000.000000 | 0.995767E 00 | -0.428882E-14 | 0.106322E 00 | -0.248948E-02 |
| 1541.000000 | 0.000000 | -1000.000000 | 0.995480E 00 | -0.154246E-13 | 0.105904E 00 | -0.191680E-02 |
| 1542.000000 | 0.000000 | -1000.000000 | 0.995381E 00 | -0.909864E-14 | 0.105766E 00 | -0.171864E-02 |
| 1543.000000 | 0.000000 | -1000.000000 | 0.995216E 00 | -0.154823E-13 | 0.105648E 00 | -0.138727E-02 |
| 1544.000000 | 0.000000 | -1000.000000 | 0.995167E 00 | -0.264079E-13 | 0.105424E 00 | -0.129014E-02 |
| 1545.000000 | 0.000000 | -1000.000000 | 0.995096E 00 | -0.252612E-13 | 0.105376E 00 | -0.114732E-02 |
| 1546.000000 | 0.000000 | -1000.000000 | 0.995021E 00 | -0.272281E-13 | 0.105258E 00 | -0.997221E-03 |
| 1547.000000 | 0.000000 | -1000.000000 | 0.994966E 00 | -0.306369E-13 | 0.105162E 00 | -0.888878E-03 |
| 1548.000000 | 0.000000 | -1000.000000 | 0.994932E 00 | -0.336086E-13 | 0.105153E 00 | -0.819630E-03 |
| 1549.000000 | 0.000000 | -1000.000000 | 0.994899E 00 | -0.292328E-13 | 0.105104E 00 | -0.753283E-03 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1550.000000 | 0.000000 | -1000.000000 | 0.994834E 00 | -0.377943E-13 | 0.105097E 00 | -0.623869E-03 |
| 1551.000000 | 0.000000 | -1000.000000 | 0.994845E 00 | -0.336489E-13 | 0.105035E 00 | -0.645861E-03 |
| 1552.000000 | 0.000000 | -1000.000000 | 0.994823E 00 | -0.307934E-13 | 0.104976E 00 | -0.601523E-03 |
| 1553.000000 | 0.000000 | -1000.000000 | 0.994807E 00 | -0.304091E-13 | 0.104914E 00 | -0.571202E-03 |
| 1554.000000 | 0.000000 | -1000.000000 | 0.994787E 00 | -0.298333E-13 | 0.104928E 00 | -0.530887E-03 |
| 1555.000000 | 0.000000 | -1000.000000 | 0.994774E 00 | -0.343503E-13 | 0.104916E 00 | -0.505182E-03 |
| 1556.000000 | 0.000000 | -1000.000000 | 0.994778E 00 | -0.303235E-13 | 0.104808E 00 | -0.512955E-03 |
| 1557.000000 | 0.000000 | -1000.000000 | 0.994760E 00 | -0.299211E-13 | 0.104826E 00 | -0.475967E-03 |
| 1558.000000 | 0.000000 | -1000.000000 | 0.994714E 00 | -0.275570E-13 | 0.104858E 00 | -0.384374E-03 |
| 1559.000000 | 0.000000 | -1000.000000 | 0.994697E 00 | -0.280286E-13 | 0.104854E 00 | -0.349372E-03 |

STREAMLINES

INITIAL POINT X= 28.00000 Y= 5.00000 Z= 0.25000

INITIAL STEPSIZE= 0.10000

ACTUAL POINTS DETERMINED BY INTEGRATOR

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 19.887316 | 5.402700 | -0.668064 | 0.995046 | 0.006203 | 0.110184 | -0.001049 |
| 20.284877 | 5.404234 | -0.624006 | 0.997193 | 0.002872 | 0.110724 | -0.005343 |
| 20.682416 | 5.405826 | -0.579681 | 0.993749 | 0.007922 | 0.111414 | 0.001545 |
| 21.079837 | 5.411845 | -0.534778 | 0.984898 | 0.021389 | 0.112272 | 0.019249 |
| 21.477072 | 5.422244 | -0.489023 | 0.981797 | 0.026645 | 0.114327 | 0.025451 |
| 21.874228 | 5.431400 | -0.442205 | 0.989436 | 0.015709 | 0.117955 | 0.010172 |
| 22.271331 | 5.433238 | -0.394316 | 1.001270 | -0.002067 | 0.122391 | -0.013456 |
| 22.668275 | 5.432495 | -0.344974 | 1.000979 | -0.002073 | 0.126674 | -0.012915 |
| 23.064974 | 5.431330 | -0.293712 | 1.002050 | -0.004345 | 0.132302 | -0.015056 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 23.461350 | 5.428754 | -0.240064 | 1.004213 | -0.008811 | 0.139552 | -0.019383 |
| 23.857268 | 5.424185 | -0.183278 | 1.005828 | -0.013411 | 0.149446 | -0.022612 |
| 24.252515 | 5.418257 | -0.122173 | 1.005584 | -0.017103 | 0.162943 | -0.022124 |
| 24.646799 | 5.409837 | -0.055421 | 1.008895 | -0.030569 | 0.179906 | -0.028747 |
| 25.039659 | 5.391396 | 0.017779 | 1.023872 | -0.071883 | 0.192456 | -0.058700 |
| 25.431363 | 5.351877 | 0.088136 | 1.049555 | -0.137490 | 0.171942 | -0.110067 |
| 25.823183 | 5.292903 | 0.142454 | 1.066037 | -0.173870 | 0.119780 | -0.143030 |
| 26.216382 | 5.228767 | 0.177949 | 1.066257 | -0.168952 | 0.077428 | -0.143471 |
| 26.413615 | 5.198322 | 0.191037 | 1.062236 | -0.159172 | 0.064593 | -0.135429 |
| 26.611240 | 5.169491 | 0.202193 | 1.057136 | -0.147844 | 0.055496 | -0.125227 |
| 26.809205 | 5.142963 | 0.211942 | 1.052785 | -0.137822 | 0.048694 | -0.116527 |
| 27.007400 | 5.117588 | 0.220588 | 1.050675 | -0.131857 | 0.043156 | -0.112306 |
| 27.205743 | 5.093065 | 0.228229 | 1.050377 | -0.128190 | 0.037817 | -0.111710 |
| 27.404185 | 5.069060 | 0.234885 | 1.051292 | -0.126303 | 0.032715 | -0.113541 |
| 27.503431 | 5.057170 | 0.237857 | 1.051910 | -0.125638 | 0.030287 | -0.114777 |
| 27.602694 | 5.045361 | 0.240608 | 1.052205 | -0.124555 | 0.028092 | -0.115367 |
| 27.701977 | 5.033690 | 0.243170 | 1.051936 | -0.122674 | 0.026261 | -0.114829 |
| 27.801287 | 5.022217 | 0.245573 | 1.051235 | -0.120221 | 0.024695 | -0.113427 |
| 27.900628 | 5.010980 | 0.247841 | 1.050308 | -0.117450 | 0.023324 | -0.111572 |
| 28.000000 | 5.000000 | 0.250000 | 1.049007 | -0.114446 | 0.022308 | -0.108970 |
| 28.099404 | 4.989298 | 0.252073 | 1.047557 | -0.111229 | 0.021443 | -0.106071 |
| 28.198841 | 4.978891 | 0.254075 | 1.045977 | -0.107864 | 0.020750 | -0.102910 |
| 28.298309 | 4.968782 | 0.256021 | 1.044426 | -0.104585 | 0.020165 | -0.099808 |
| 28.397807 | 4.958959 | 0.257917 | 1.043027 | -0.101553 | 0.019624 | -0.097010 |
| 28.497331 | 4.949388 | 0.259761 | 1.041993 | -0.099079 | 0.018988 | -0.094941 |
| 28.596876 | 4.940024 | 0.261546 | 1.041130 | -0.096895 | 0.018377 | -0.093215 |
| 28.796026 | 4.921528 | 0.264962 | 1.037990 | -0.090486 | 0.017508 | -0.086936 |
| 28.995301 | 4.905236 | 0.268257 | 1.034521 | -0.083666 | 0.016703 | -0.079997 |
| 29.194665 | 4.889674 | 0.271380 | 1.032016 | -0.078192 | 0.015662 | -0.074987 |
| 29.394111 | 4.875044 | 0.274324 | 1.029269 | -0.072299 | 0.014854 | -0.069495 |
| 29.593640 | 4.861623 | 0.277149 | 1.026388 | -0.065880 | 0.014295 | -0.063732 |
| 29.793232 | 4.849149 | 0.279851 | 1.026099 | -0.063353 | 0.013356 | -0.063155 |
| 30.192528 | 4.825964 | 0.284884 | 1.023474 | -0.055961 | 0.012562 | -0.057904 |
| 30.591959 | 4.805197 | 0.289654 | 1.022061 | -0.050755 | 0.011794 | -0.055077 |
| 30.991483 | 4.786218 | 0.294090 | 1.020762 | -0.045995 | 0.010977 | -0.052480 |
| 31.391109 | 4.769498 | 0.298411 | 1.017634 | -0.038642 | 0.011385 | -0.046224 |
| 31.790848 | 4.755883 | 0.303199 | 1.013864 | -0.030568 | 0.013180 | -0.038683 |
| 32.190670 | 4.745411 | 0.308839 | 1.010218 | -0.022514 | 0.015340 | -0.031392 |
| 32.590547 | 4.737928 | 0.315315 | 1.007587 | -0.015503 | 0.017241 | -0.026130 |
| 32.990451 | 4.732927 | 0.322484 | 1.006204 | -0.009962 | 0.018788 | -0.023364 |
| 33.390357 | 4.729947 | 0.330531 | 1.004304 | -0.005071 | 0.022156 | -0.019565 |
| 33.790241 | 4.728807 | 0.340093 | 1.002548 | -0.000726 | 0.025719 | -0.016053 |
| 34.190090 | 4.729292 | 0.351065 | 1.000854 | 0.003059 | 0.029224 | -0.012665 |
| 34.589895 | 4.731079 | 0.363413 | 0.999623 | 0.005627 | 0.032510 | -0.010202 |
| 34.989656 | 4.733593 | 0.377012 | 0.999362 | 0.006662 | 0.035403 | -0.009681 |
| 35.389378 | 4.736258 | 0.391682 | 0.999874 | 0.006489 | 0.037902 | -0.010705 |
| 36.188734 | 4.739738 | 0.423579 | 1.002912 | 0.002524 | 0.041966 | -0.016781 |
| 36.987971 | 4.741210 | 0.458446 | 1.003994 | 0.001667 | 0.045449 | -0.018944 |
| 37.787099 | 4.742117 | 0.495797 | 1.005117 | 0.000308 | 0.048497 | -0.021190 |
| 38.586117 | 4.741710 | 0.535409 | 1.006283 | -0.001370 | 0.051155 | -0.023522 |
| 39.385040 | 4.740098 | 0.576863 | 1.007677 | -0.002540 | 0.053226 | -0.026311 |
| 40.183876 | 4.738626 | 0.619970 | 1.006302 | -0.000392 | 0.055618 | -0.023559 |

INITIAL POINT X= 28.00000 Y= 5.00000 Z= -0.50000

INITIAL STEPSIZE= 0.20000

ACTUAL POINTS DETERMINED BY INTEGRATOR

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| 19.698917 | 4.025370 | -1.139750 | 0.972324 | 0.048993 | 0.105959 | 0.044396 |
| 20.096007 | 4.044373 | -1.095409 | 0.979819 | 0.041466 | 0.113158 | 0.029407 |
| 20.492882 | 4.057620 | -1.047399 | 0.990244 | 0.029943 | 0.123197 | 0.008556 |
| 20.889555 | 4.071875 | -0.997935 | 0.984249 | 0.041377 | 0.123574 | 0.020546 |
| 21.285893 | 4.091317 | -0.947604 | 0.979215 | 0.053890 | 0.126354 | 0.030613 |
| 21.681760 | 4.115202 | -0.895503 | 0.976105 | 0.065297 | 0.130097 | 0.036833 |
| 22.077143 | 4.145166 | -0.842838 | 0.968244 | 0.083765 | 0.127958 | 0.052556 |
| 22.471904 | 4.184523 | -0.791738 | 0.956446 | 0.108908 | 0.120303 | 0.076153 |
| 22.865814 | 4.235175 | -0.744226 | 0.944605 | 0.134197 | 0.108785 | 0.099835 |
| 23.258922 | 4.295291 | -0.701279 | 0.936667 | 0.152540 | 0.096284 | 0.115309 |
| 23.651492 | 4.362117 | -0.663647 | 0.931889 | 0.163437 | 0.082754 | 0.125266 |
| 24.044130 | 4.431424 | -0.631578 | 0.931378 | 0.164016 | 0.069720 | 0.126287 |
| 24.437351 | 4.499592 | -0.604574 | 0.934768 | 0.158168 | 0.058890 | 0.119508 |
| 24.831480 | 4.563946 | -0.581788 | 0.939977 | 0.148597 | 0.049201 | 0.109090 |
| 25.226391 | 4.624811 | -0.563404 | 0.942073 | 0.142464 | 0.038778 | 0.104898 |
| 25.621729 | 4.683964 | -0.549000 | 0.941836 | 0.140588 | 0.030109 | 0.105372 |
| 26.017152 | 4.743250 | -0.537799 | 0.940027 | 0.141896 | 0.023517 | 0.108990 |
| 26.412621 | 4.802608 | -0.528868 | 0.940777 | 0.138787 | 0.019494 | 0.107490 |
| 26.808545 | 4.859032 | -0.521158 | 0.944713 | 0.129047 | 0.017444 | 0.099618 |
| 27.006761 | 4.885435 | -0.517548 | 0.947702 | 0.122854 | 0.017103 | 0.093641 |
| 27.205162 | 4.910416 | -0.513976 | 0.951136 | 0.116384 | 0.017088 | 0.086772 |
| 27.403725 | 4.934081 | -0.510419 | 0.953946 | 0.110950 | 0.017033 | 0.081152 |
| 27.602408 | 4.956720 | -0.506884 | 0.955846 | 0.106956 | 0.016894 | 0.077352 |
| 27.801173 | 4.978641 | -0.503395 | 0.956993 | 0.104115 | 0.016624 | 0.075057 |
| 28.000000 | 5.000000 | -0.500000 | 0.957483 | 0.101606 | 0.016038 | 0.074079 |
| 28.198881 | 5.020872 | -0.496736 | 0.957520 | 0.099424 | 0.015392 | 0.074004 |
| 28.397811 | 5.041294 | -0.493601 | 0.957585 | 0.097002 | 0.014813 | 0.073873 |
| 28.596803 | 5.061118 | -0.490560 | 0.958377 | 0.093736 | 0.014542 | 0.072290 |
| 28.795869 | 5.080184 | -0.487562 | 0.959682 | 0.089885 | 0.014336 | 0.069681 |
| 28.995004 | 5.098526 | -0.484622 | 0.960098 | 0.087344 | 0.013966 | 0.068847 |
| 29.194177 | 5.116470 | -0.481774 | 0.959772 | 0.085768 | 0.013466 | 0.069499 |
| 29.592664 | 5.150803 | -0.476308 | 0.960789 | 0.079901 | 0.012833 | 0.067466 |
| 29.991328 | 5.183051 | -0.471207 | 0.960474 | 0.075856 | 0.011568 | 0.068095 |
| 30.390194 | 5.212810 | -0.466816 | 0.963466 | 0.066898 | 0.009553 | 0.062113 |
| 30.789337 | 5.238760 | -0.463257 | 0.965668 | 0.059332 | 0.007663 | 0.057708 |
| 31.188674 | 5.261567 | -0.460442 | 0.969282 | 0.050747 | 0.006109 | 0.050479 |
| 31.588210 | 5.280680 | -0.458255 | 0.973076 | 0.042319 | 0.004438 | 0.042891 |
| 31.987882 | 5.296790 | -0.456678 | 0.975138 | 0.036995 | 0.003399 | 0.038767 |
| 32.387618 | 5.311285 | -0.455617 | 0.975500 | 0.034084 | 0.001561 | 0.038044 |
| 32.787406 | 5.324227 | -0.455300 | 0.977796 | 0.028089 | 0.000183 | 0.033451 |
| 33.187274 | 5.334505 | -0.455332 | 0.979696 | 0.022754 | -0.000203 | 0.029652 |
| 33.587193 | 5.342533 | -0.455505 | 0.981920 | 0.016208 | -0.000704 | 0.025203 |
| 33.987158 | 5.347711 | -0.455695 | 0.984105 | 0.009251 | 0.000063 | 0.020834 |
| 34.387151 | 5.349948 | -0.455357 | 0.986774 | 0.001643 | 0.001761 | 0.015496 |
| 34.787145 | 5.348944 | -0.453877 | 0.989527 | -0.006780 | 0.006123 | 0.009990 |
| 35.187107 | 5.344974 | -0.450102 | 0.990841 | -0.011946 | 0.012933 | 0.007362 |
| 35.587017 | 5.339479 | -0.443612 | 0.991611 | -0.015048 | 0.018927 | 0.005822 |
| 35.986866 | 5.332818 | -0.434919 | 0.992885 | -0.018256 | 0.024059 | 0.003274 |
| 36.386639 | 5.324582 | -0.424298 | 0.994806 | -0.023089 | 0.028687 | -0.000569 |
| 36.786309 | 5.314158 | -0.411884 | 0.996655 | -0.028974 | 0.033140 | -0.004266 |
| 37.185852 | 5.301389 | -0.397661 | 0.998029 | -0.034634 | 0.037968 | -0.007014 |
| 37.984445 | 5.270024 | -0.362308 | 1.000213 | -0.043872 | 0.051986 | -0.011382 |
| 38.781859 | 5.232774 | -0.310416 | 1.002077 | -0.045380 | 0.082024 | -0.015110 |
| 39.577878 | 5.210860 | -0.234574 | 1.003691 | -0.005481 | 0.099092 | -0.018339 |
| 40.374514 | 5.219806 | -0.161366 | 1.005040 | 0.012659 | 0.079237 | -0.021037 |

INITIAL POINT X= 15.40000 Y= 1.50000 Z= 0.00000

INITIAL STEPSIZE= 0.10000

ACTUAL POINTS DETERMINED BY INTEGRATOR

| X | Y | Z | U | V | W | CP |
|---|---|---|---|---|---|---|
| -0.089190 | 0.840364 | -1.598071 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 0.308619 | 0.840364 | -1.556260 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 0.706428 | 0.840364 | -1.514448 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 1.104237 | 0.840364 | -1.472637 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 1.502045 | 0.840364 | -1.430826 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 1.899892 | 0.840692 | -1.389466 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 2.297559 | 0.839562 | -1.346055 | 0.994522 | 0.000000 | 0.104528 | -0.000000 |
| 2.695897 | 0.843815 | -1.310216 | 0.972407 | 0.019191 | 0.075499 | 0.044229 |
| 3.094768 | 0.854107 | -1.282155 | 0.963962 | 0.029581 | 0.062696 | 0.061119 |
| 3.493754 | 0.867528 | -1.257040 | 0.964660 | 0.033271 | 0.061395 | 0.059723 |
| 3.892621 | 0.880602 | -1.229984 | 0.971772 | 0.031279 | 0.068455 | 0.045500 |
| 4.291407 | 0.893838 | -1.201745 | 0.971996 | 0.034723 | 0.067595 | 0.045052 |
| 4.690212 | 0.909987 | -1.175472 | 0.969810 | 0.041835 | 0.062712 | 0.051425 |
| 5.088949 | 0.927192 | -1.148772 | 0.972684 | 0.042617 | 0.066529 | 0.043676 |
| 5.487598 | 0.945317 | -1.121370 | 0.973003 | 0.046507 | 0.066657 | 0.043037 |
| 5.886171 | 0.965649 | -1.094442 | 0.972109 | 0.051831 | 0.065717 | 0.044825 |
| 6.284634 | 0.987338 | -1.066940 | 0.974093 | 0.054329 | 0.068432 | 0.040859 |
| 6.682970 | 1.010212 | -1.038556 | 0.974828 | 0.057915 | 0.070238 | 0.039388 |
| 7.081162 | 1.034818 | -1.009632 | 0.974927 | 0.062150 | 0.071805 | 0.039189 |
| 7.479195 | 1.060734 | -0.979659 | 0.976313 | 0.064867 | 0.075186 | 0.036419 |
| 7.877052 | 1.087671 | -0.948296 | 0.977386 | 0.067630 | 0.078720 | 0.034272 |
| 8.274716 | 1.115869 | -0.915622 | 0.977906 | 0.070761 | 0.082164 | 0.033232 |
| 8.672175 | 1.145045 | -0.881345 | 0.979147 | 0.072727 | 0.086735 | 0.030750 |
| 9.069422 | 1.174792 | -0.845148 | 0.980452 | 0.074150 | 0.091769 | 0.028140 |
| 9.466441 | 1.205154 | -0.807023 | 0.981233 | 0.075730 | 0.096708 | 0.026578 |
| 9.863225 | 1.235946 | -0.766846 | 0.982339 | 0.076394 | 0.102238 | 0.024365 |
| 10.259774 | 1.266690 | -0.724386 | 0.983753 | 0.076050 | 0.108262 | 0.021538 |
| 10.656085 | 1.297210 | -0.679602 | 0.984822 | 0.075498 | 0.114207 | 0.019399 |
| 11.052154 | 1.327364 | -0.632491 | 0.985918 | 0.074251 | 0.120275 | 0.017208 |
| 11.447992 | 1.356667 | -0.582961 | 0.987445 | 0.071703 | 0.126635 | 0.014153 |
| 11.843608 | 1.384740 | -0.531010 | 0.988836 | 0.068589 | 0.132829 | 0.011372 |
| 12.239006 | 1.411482 | -0.476746 | 0.989943 | 0.065151 | 0.138692 | 0.009157 |
| 12.634202 | 1.436554 | -0.420258 | 0.991529 | 0.060316 | 0.144505 | 0.005986 |
| 13.029220 | 1.455403 | -0.361626 | 0.993333 | 0.054423 | 0.150025 | 0.002377 |
| 13.226667 | 1.469931 | -0.331564 | 0.994010 | 0.051542 | 0.152528 | 0.001024 |
| 13.424075 | 1.479875 | -0.301046 | 0.994592 | 0.048642 | 0.154871 | -0.000140 |
| 13.621445 | 1.489238 | -0.270103 | 0.995110 | 0.045689 | 0.157056 | -0.001177 |
| 13.818780 | 1.497975 | -0.238759 | 0.995771 | 0.042387 | 0.159131 | -0.002498 |
| 14.016088 | 1.505951 | -0.207053 | 0.996529 | 0.038803 | 0.161072 | -0.004015 |
| 14.213369 | 1.513199 | -0.175024 | 0.997342 | 0.035008 | 0.162860 | -0.005641 |
| 14.410638 | 1.519599 | -0.142714 | 1.000904 | 0.027122 | 0.163348 | -0.012764 |
| 14.608127 | 1.521562 | -0.111232 | 1.017068 | -0.003438 | 0.160088 | -0.045093 |
| 14.805762 | 1.518650 | -0.080738 | 1.025004 | -0.025495 | 0.153498 | -0.060965 |
| 14.904664 | 1.515839 | -0.066231 | 1.026714 | -0.032067 | 0.147403 | -0.064385 |
| 15.003638 | 1.512594 | -0.052320 | 1.026690 | -0.034725 | 0.141428 | -0.064336 |
| 15.102678 | 1.509230 | -0.038908 | 1.025505 | -0.034684 | 0.136792 | -0.061967 |
| 15.201765 | 1.505942 | -0.025833 | 1.023627 | -0.033129 | 0.133899 | -0.058210 |
| 15.300879 | 1.502838 | -0.012925 | 1.021328 | -0.030833 | 0.132693 | -0.053612 |
| 15.400000 | 1.500000 | 0.000000 | 1.016966 | -0.026069 | 0.134562 | -0.044889 |
| 15.499096 | 1.497618 | 0.013199 | 1.014519 | -0.023337 | 0.135663 | -0.039995 |
| 15.598188 | 1.495367 | 0.026457 | 1.014064 | -0.023005 | 0.135587 | -0.039084 |
| 15.697285 | 1.493081 | 0.039672 | 1.014674 | -0.023956 | 0.134840 | -0.040304 |
| 15.796393 | 1.490677 | 0.052777 | 1.015682 | -0.025280 | 0.133558 | -0.042321 |
| 15.895519 | 1.488143 | 0.065722 | 1.016871 | -0.026673 | 0.131795 | -0.044698 |

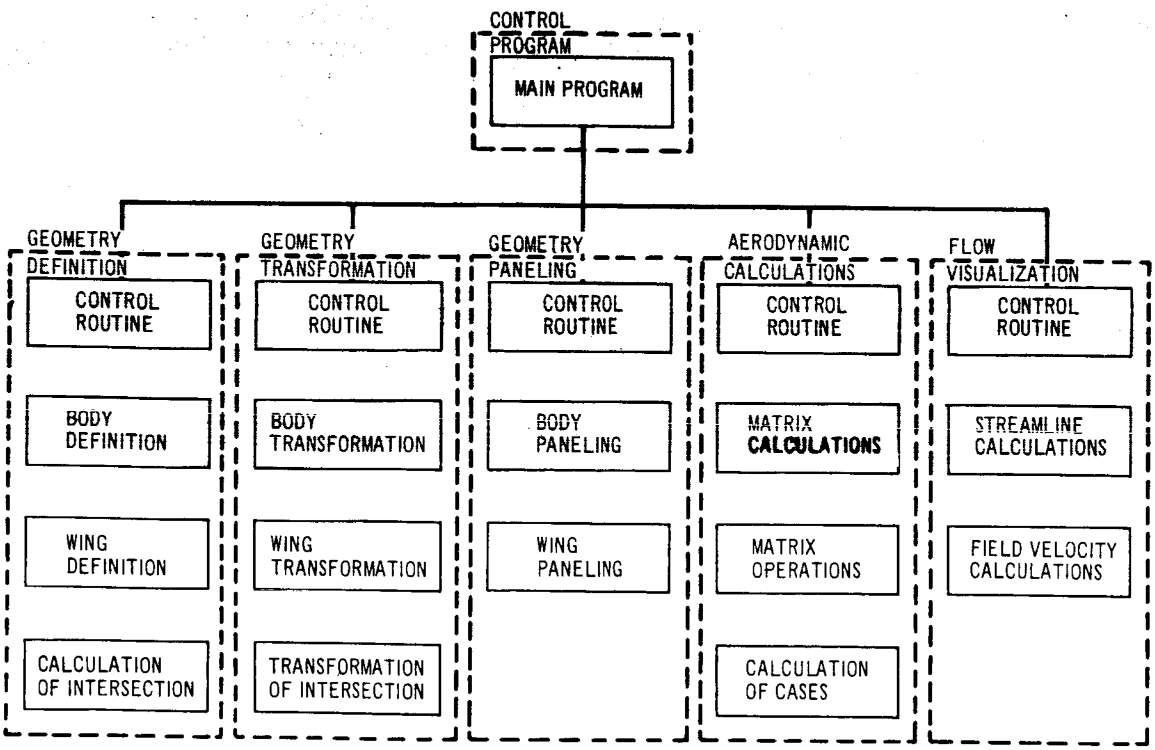
| | | | | | | |
|---|---|---|---|---|---|---|
| 15.994669 | 1.485474 | 0.078460 | 1.018230 | -0.028116 | 0.129585 | -0.047417 |
| 16.193055 | 1.479737 | 0.103151 | 1.021235 | -0.030778 | 0.123998 | -0.053426 |
| 16.391582 | 1.473560 | 0.126575 | 1.024361 | -0.032714 | 0.117182 | -0.059678 |
| 16.590246 | 1.467325 | 0.148799 | 1.024946 | -0.031178 | 0.112284 | -0.060847 |
| 16.789016 | 1.461563 | 0.170182 | 1.024592 | -0.028083 | 0.108196 | -0.060120 |
| 16.987880 | 1.456476 | 0.190847 | 1.023709 | -0.024276 | 0.104762 | -0.058373 |
| 17.186825 | 1.452170 | 0.210904 | 1.022794 | -0.019955 | 0.101553 | -0.056545 |
| 17.385858 | 1.448748 | 0.230241 | 1.024301 | -0.015144 | 0.096861 | -0.059558 |
| 17.784236 | 1.444854 | 0.265984 | 1.028934 | -0.004837 | 0.086911 | -0.068825 |
| 18.183013 | 1.444639 | 0.297174 | 1.035921 | 0.006061 | 0.074728 | -0.082798 |
| 18.582051 | 1.449404 | 0.324441 | 1.034550 | 0.018466 | 0.067786 | -0.080056 |
| 18.981142 | 1.458459 | 0.349823 | 1.030407 | 0.027767 | 0.063794 | -0.071770 |
| 19.380249 | 1.470097 | 0.373851 | 1.028386 | 0.030978 | 0.059846 | -0.067729 |
| 19.779429 | 1.482481 | 0.396230 | 1.027962 | 0.032834 | 0.055137 | -0.066880 |
| 20.178690 | 1.495563 | 0.416712 | 1.027490 | 0.034633 | 0.050463 | -0.065936 |
| 20.578015 | 1.509343 | 0.435398 | 1.027267 | 0.036271 | 0.045654 | -0.065490 |
| 20.977403 | 1.523493 | 0.452383 | 1.025810 | 0.036014 | 0.042023 | -0.062575 |
| 21.376844 | 1.537880 | 0.467843 | 1.026858 | 0.038801 | 0.036734 | -0.064672 |
| 21.776325 | 1.553401 | 0.481066 | 1.028306 | 0.040574 | 0.031664 | -0.067569 |
| 22.175857 | 1.569007 | 0.492479 | 1.030095 | 0.039104 | 0.027237 | -0.071144 |
| 22.575467 | 1.583656 | 0.502302 | 1.031866 | 0.036400 | 0.023514 | -0.074689 |
| 22.975133 | 1.597566 | 0.510909 | 1.031347 | 0.036108 | 0.021219 | -0.073651 |
| 23.374807 | 1.611651 | 0.518771 | 1.029961 | 0.036626 | 0.019361 | -0.070879 |
| 23.774470 | 1.626328 | 0.526092 | 1.028192 | 0.039391 | 0.018543 | -0.067341 |
| 24.174095 | 1.642188 | 0.533035 | 1.027721 | 0.042056 | 0.016900 | -0.066397 |
| 24.573656 | 1.659410 | 0.540322 | 1.021656 | 0.046600 | 0.022313 | -0.054269 |
| 24.973075 | 1.678274 | 0.550731 | 1.014604 | 0.049122 | 0.030965 | -0.040164 |
| 25.372354 | 1.697967 | 0.564469 | 1.009718 | 0.050553 | 0.037906 | -0.030392 |
| 25.771521 | 1.718074 | 0.580614 | 1.007229 | 0.050754 | 0.043503 | -0.025415 |
| 26.170608 | 1.737994 | 0.598843 | 1.006198 | 0.049462 | 0.048389 | -0.023352 |
| 26.569635 | 1.757227 | 0.619013 | 1.005081 | 0.047420 | 0.053349 | -0.021118 |
| 26.968634 | 1.775710 | 0.640426 | 1.006429 | 0.046001 | 0.053263 | -0.023815 |
| 27.367665 | 1.793628 | 0.661711 | 1.005942 | 0.044130 | 0.055096 | -0.022841 |
| 27.766663 | 1.811021 | 0.684033 | 1.004991 | 0.044086 | 0.057257 | -0.020939 |
| 28.165595 | 1.828825 | 0.707190 | 1.003829 | 0.045875 | 0.059325 | -0.018614 |
| 28.564452 | 1.847217 | 0.731157 | 1.002969 | 0.046133 | 0.061252 | -0.016895 |
| 28.963272 | 1.865340 | 0.755937 | 1.001990 | 0.044660 | 0.063365 | -0.014937 |
| 29.362070 | 1.882749 | 0.781576 | 1.001187 | 0.042833 | 0.065359 | -0.013330 |
| 29.760839 | 1.899685 | 0.807964 | 1.000406 | 0.042603 | 0.067019 | -0.011768 |

# 4. PROGRAM RECORDS

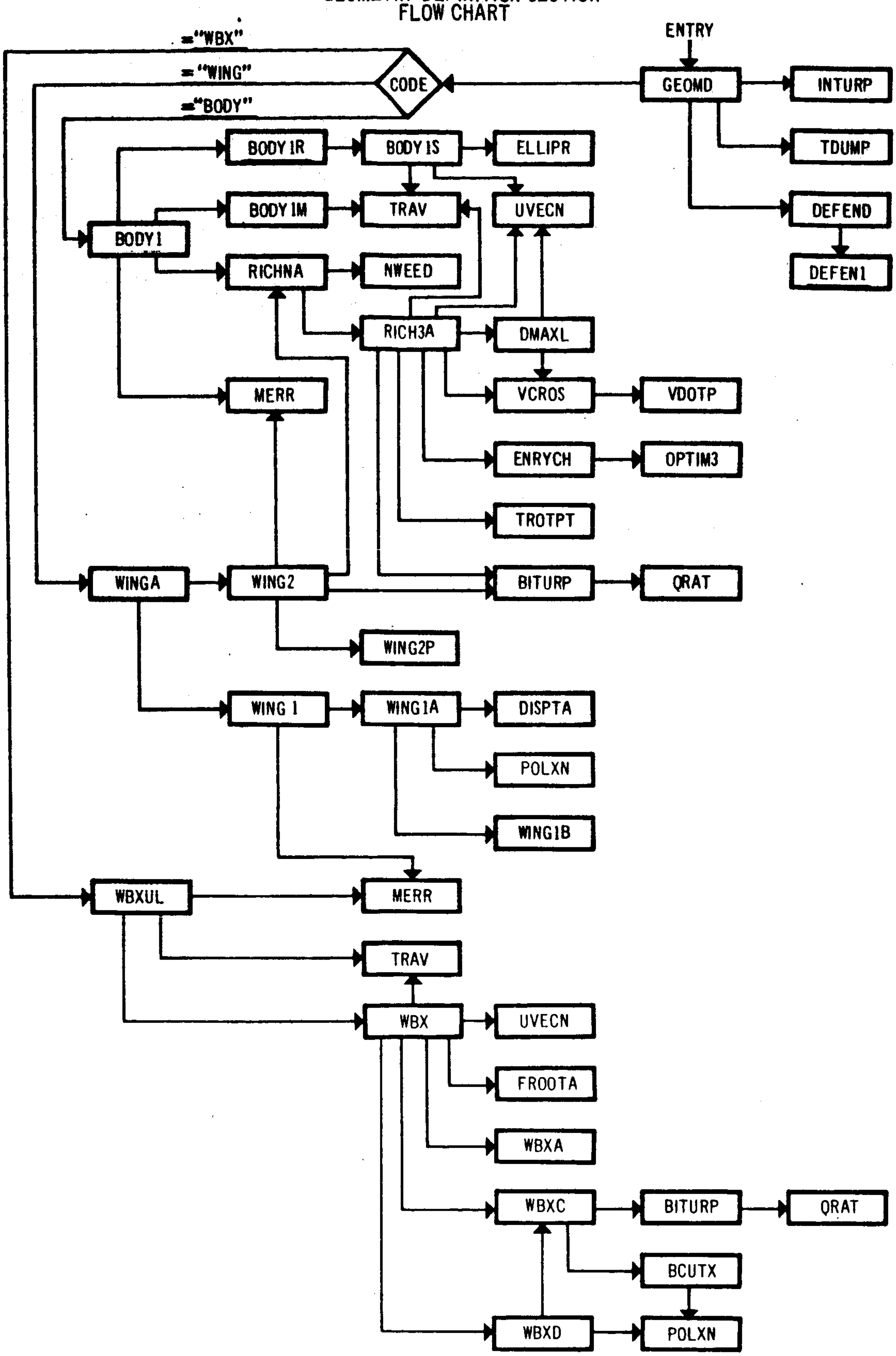
## 4.1 Flow Charts

A program flow chart, supplemented by flow charts of each of the five main program sections, and a program overlay structure diagram are presented in this section.

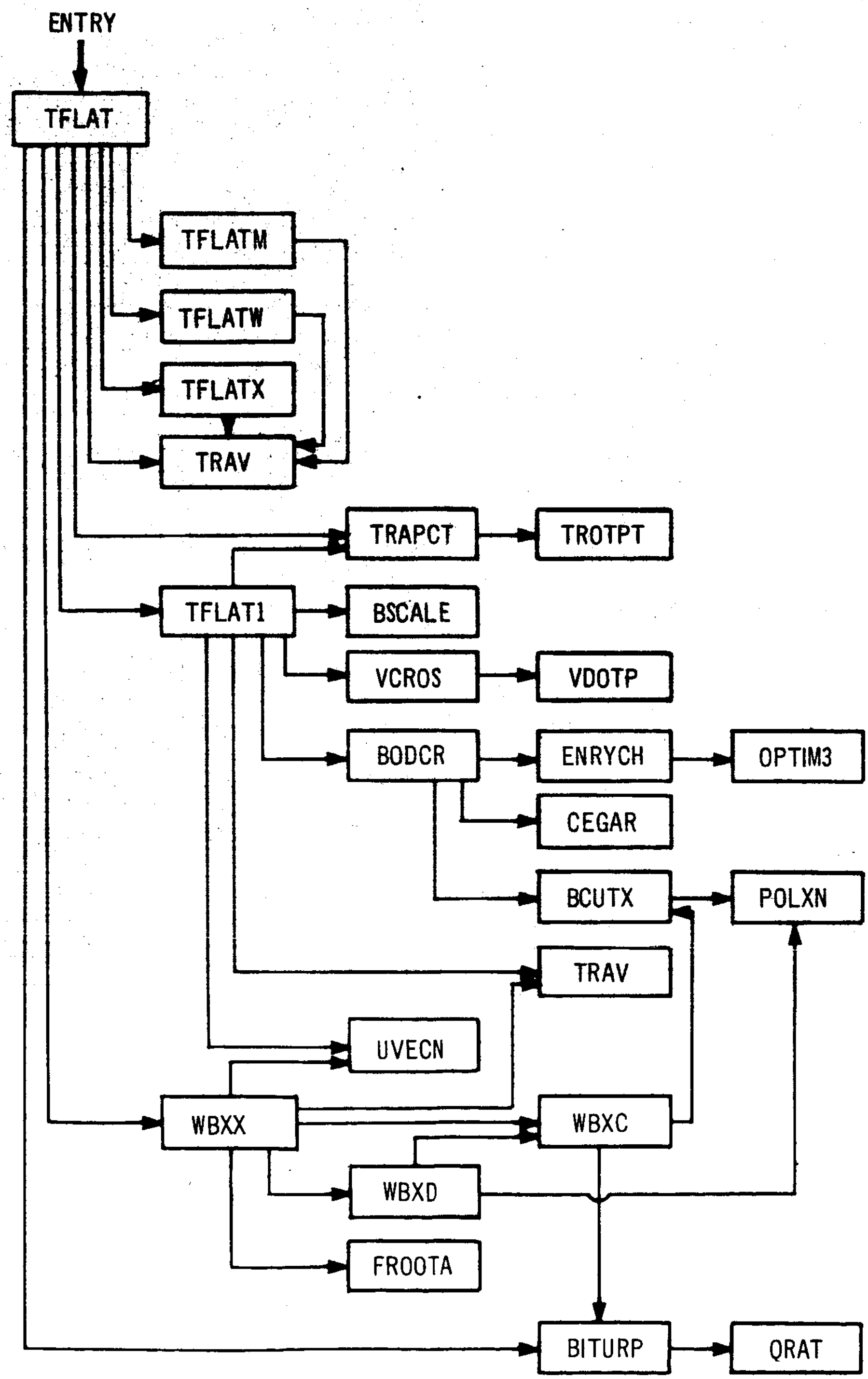
PROGRAM FLOW CHART



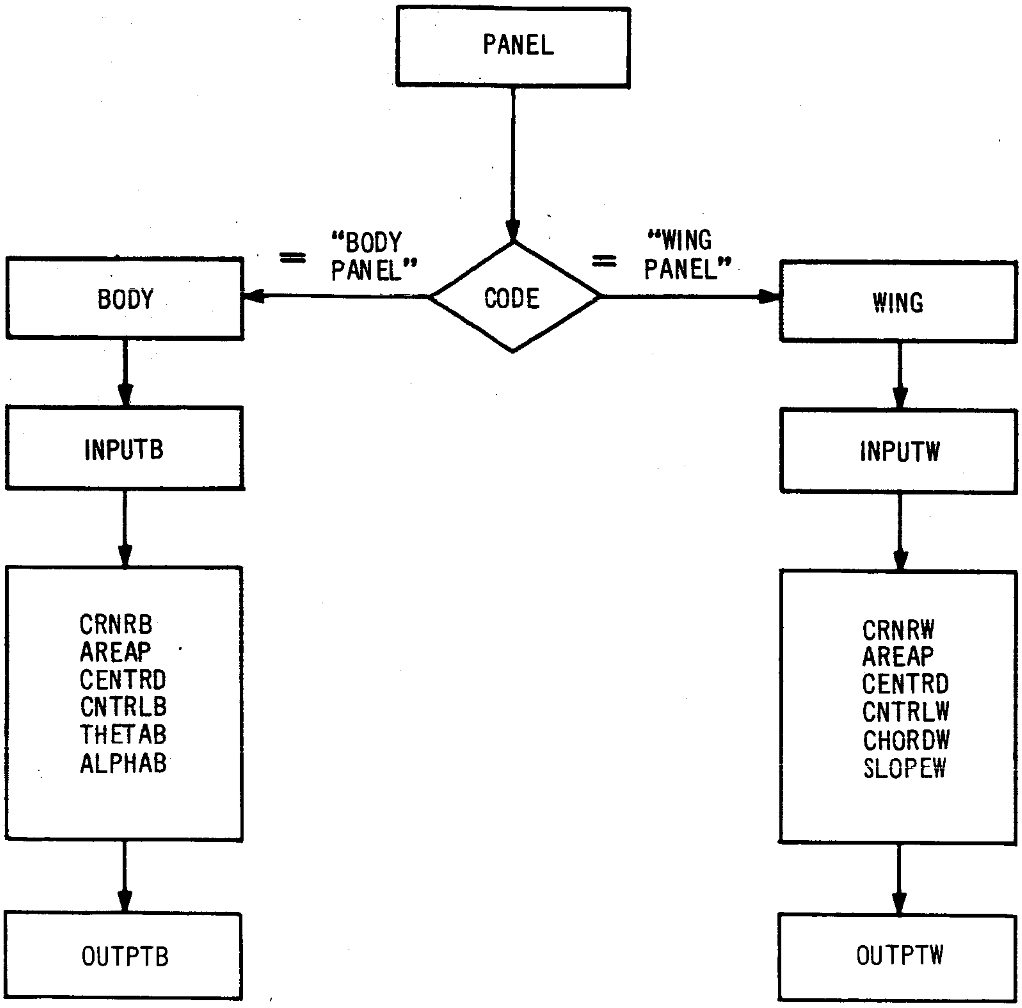
GEOMETRY DEFINITION SECTION
FLOW CHART



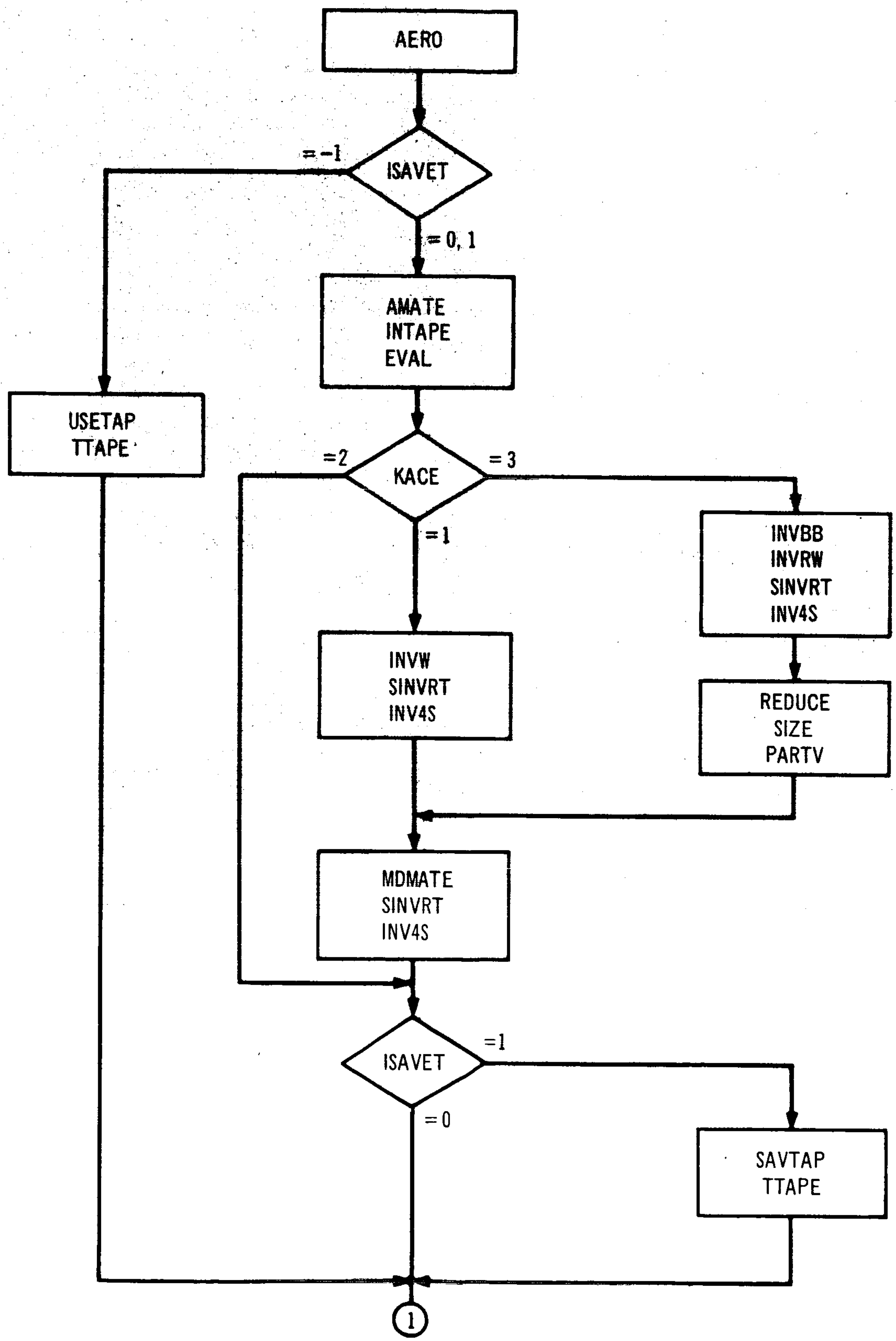
GEOMETRY TRANSFORMATION SECTION
FLOW CHART

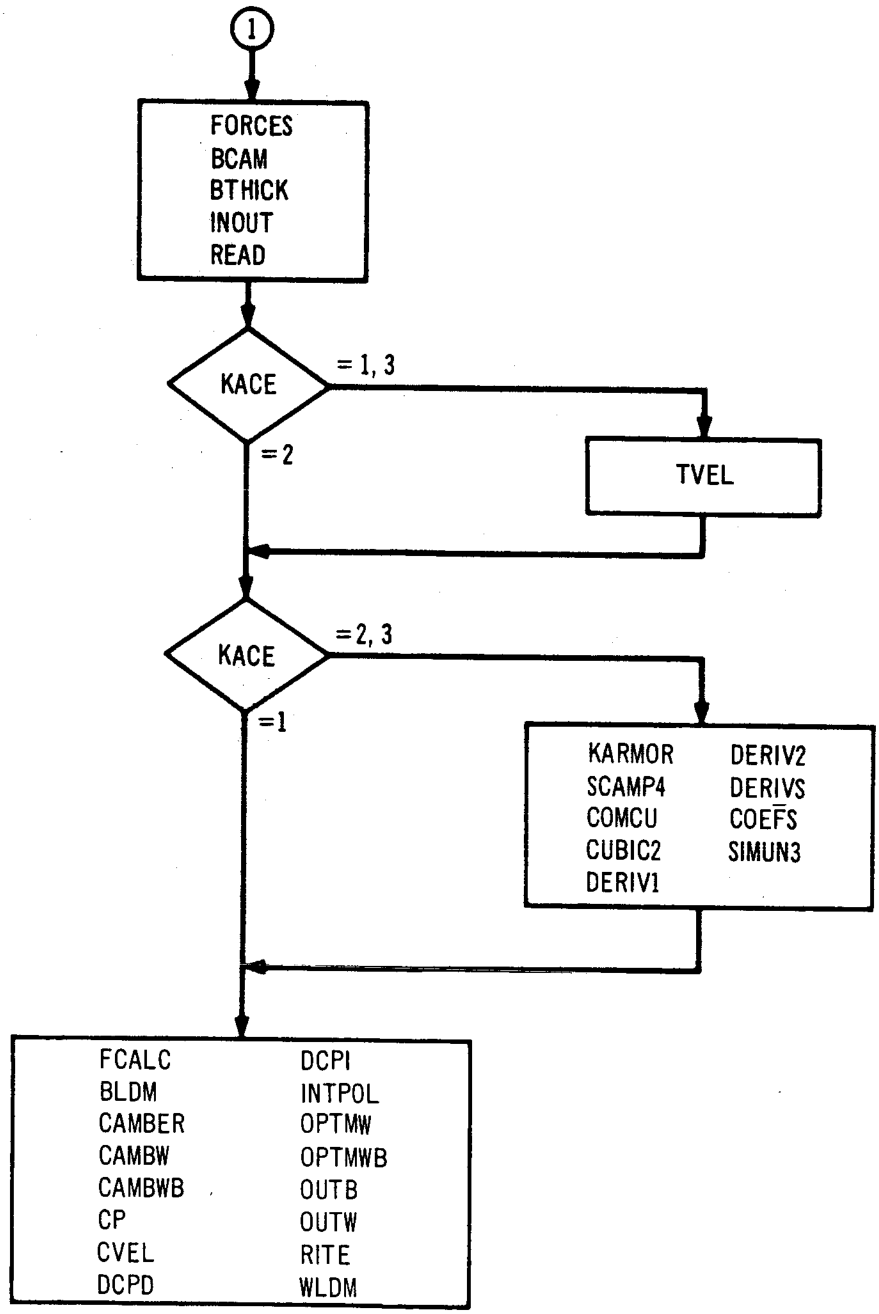


GEOMETRY PANELING SECTION
FLOW CHART

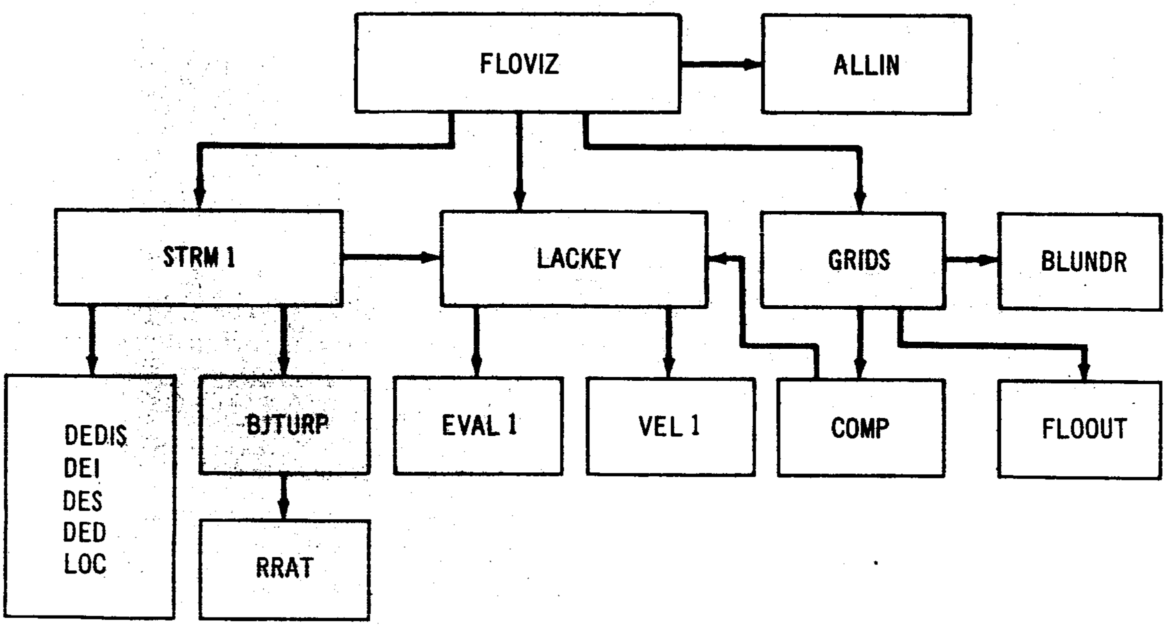


AERODYNAMIC SECTION
FLOW CHART

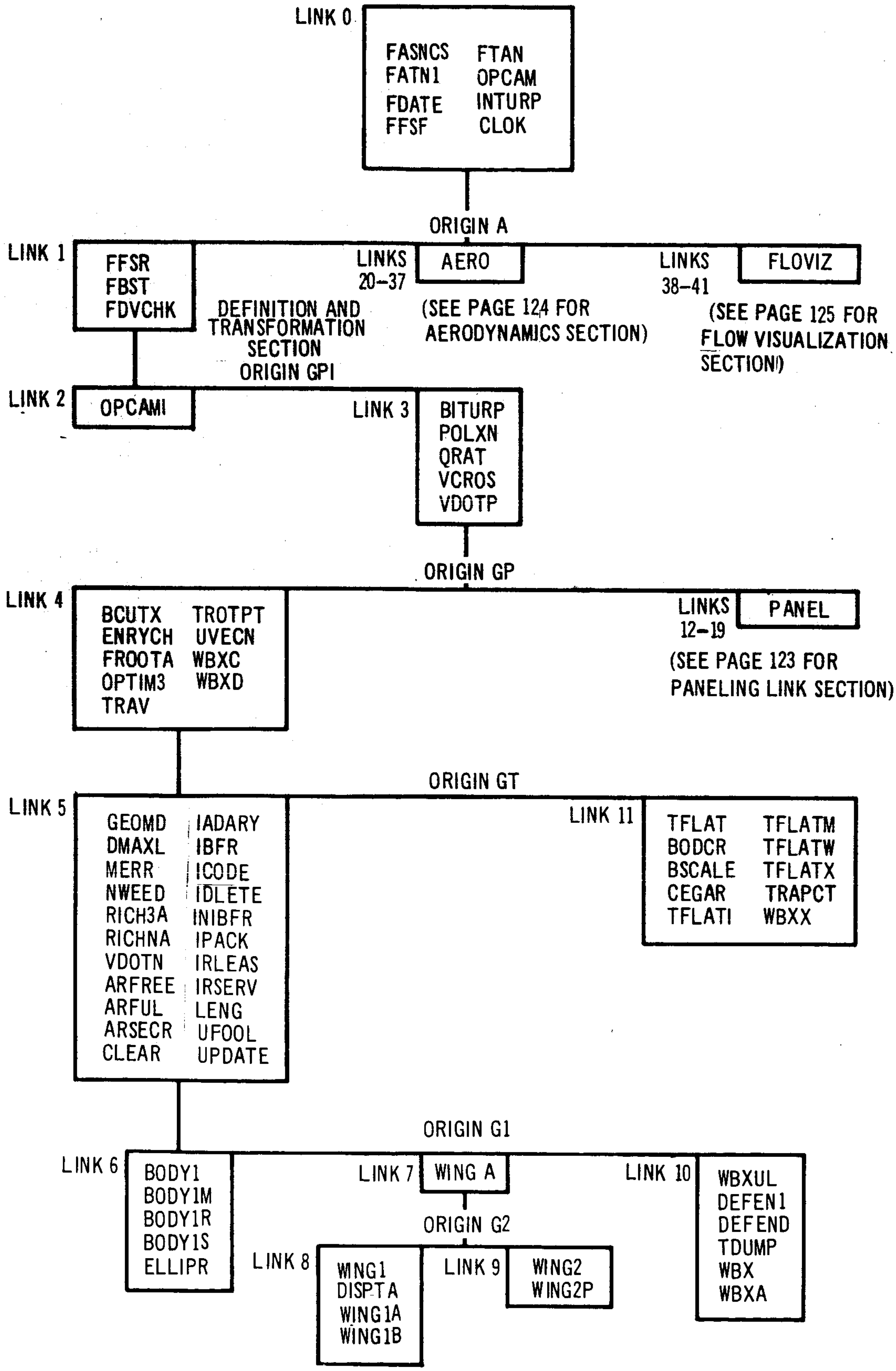



1
FORCES
BCAM
BTHICK
INOUT
READ
KACE
= 1, 3
= 2
TVEL
KACE
= 2, 3
= 1
KARMOR
SCAMP4
COMCU
CUBIC2
DERIV1
DERIV2
DERIVS
COEFS
SIMUN3
FCALC
BLDM
CAMBER
CAMBW
CAMBWB
CP
CVEL
DCPD
DCPI
INTPOL
OPTMW
OPTMWB
OUTB
OUTW
RITE
WLDM

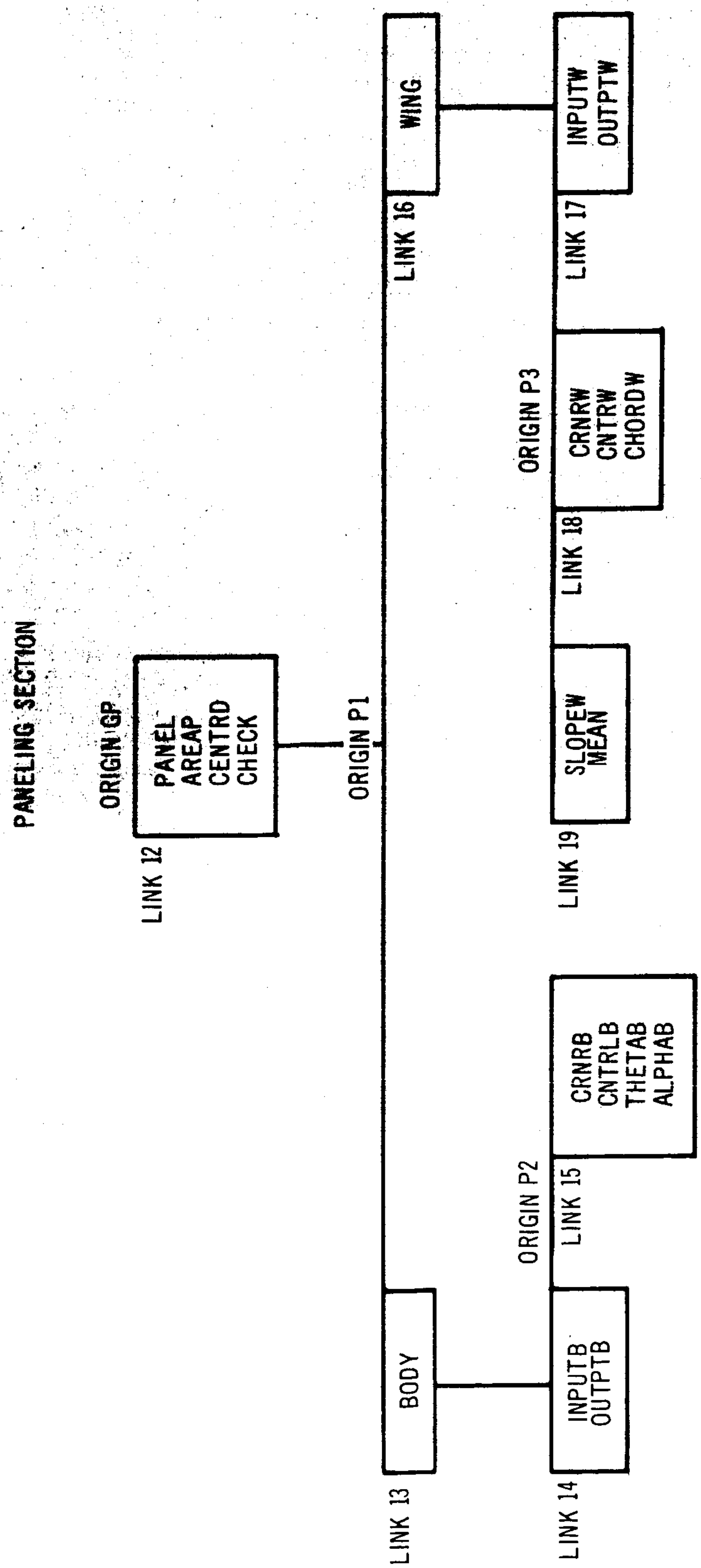
FLOW VISUALIZATION SECTION
FLOW CHART

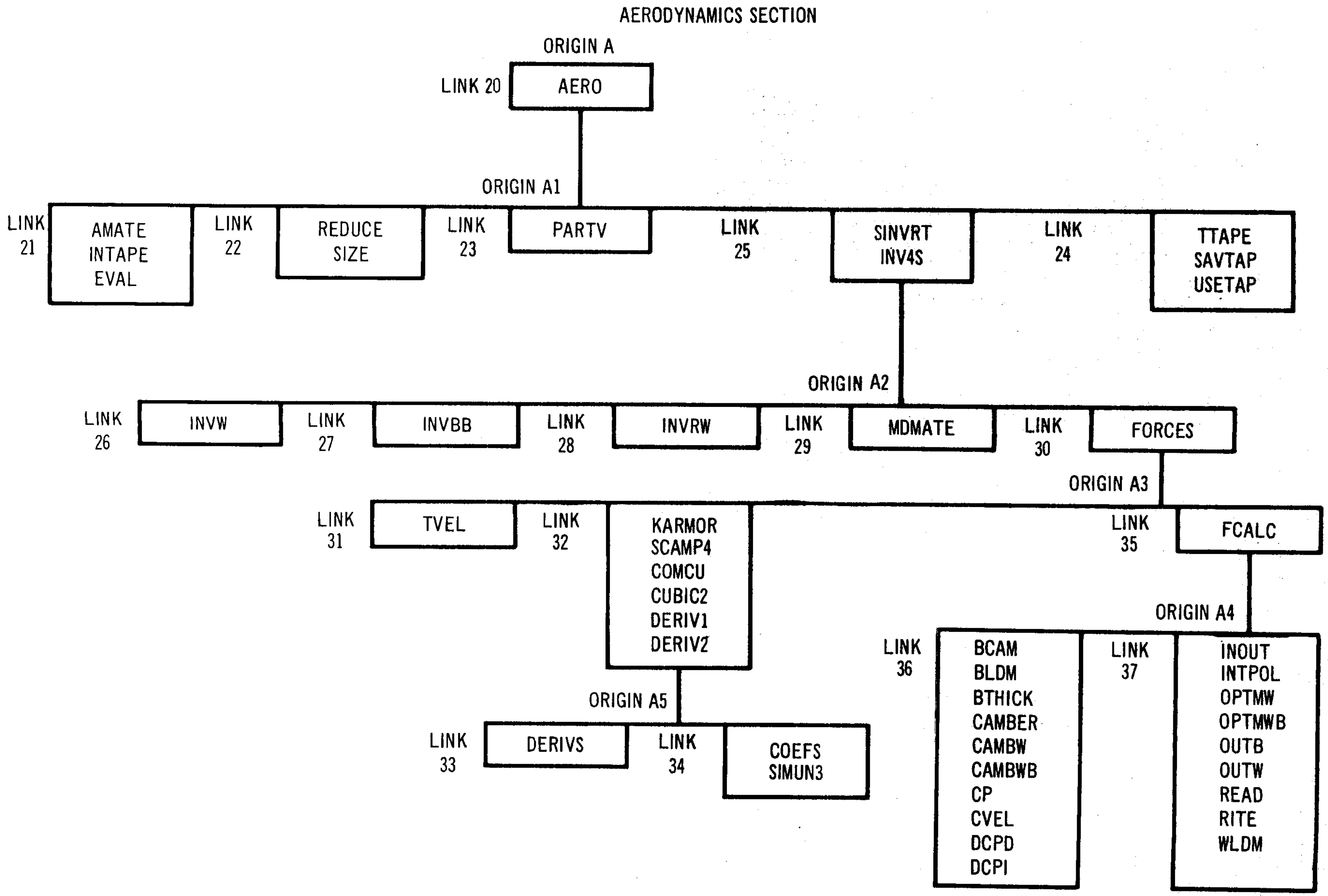


PROGRAM OVERLAY STRUCTURE

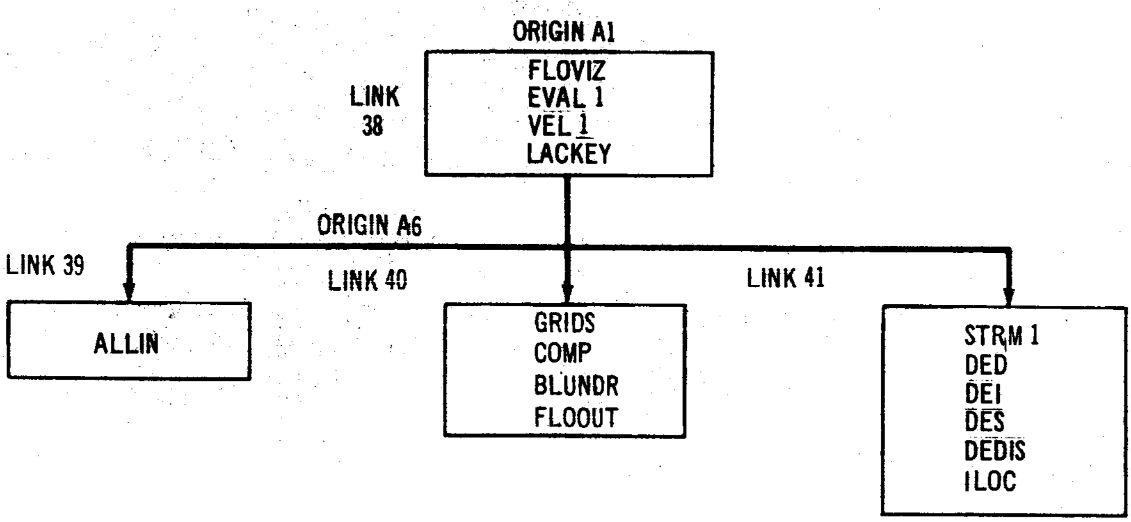


PROGRAM OVERLAY STRUCTURE (CONT)




AERODYNAMICS SECTION
ORIGIN A
LINK 20
AERO
ORIGIN A1
LINK 21
AMATE
INTAPE
EVAL
LINK 22
REDUCE
SIZE
LINK 23
PARTV
LINK 25
SINVRT
INV4S
LINK 24
TTAPE
SAVTAP
USETAP
ORIGIN A2
LINK 26
INVW
LINK 27
INVBB
LINK 28
INVRW
LINK 29
MDMATE
LINK 30
FORCES
ORIGIN A3
LINK 31
TVEL
LINK 32
KARMOR
SCAMP4
COMCU
CUBIC2
DERIV1
DERIV2
LINK 35
FCALC
ORIGIN A4
ORIGIN A5
LINK 33
DERIVS
LINK 34
COEFS
SIMUN3
LINK 36
BCAM
BLDM
BTHICK
CAMBER
CAMBW
CAMBWB
CP
CVEL
DCPD
DCPI
LINK 37
INOUT
INTPOL
OPTMW
OPTMWB
OUTB
OUTW
READ
RITE
WLDM

FLOW VISUALIZATION SECTION



## 4.2 Subroutine Descriptions

The subroutines listed alphabetically in the accompanying index (with the exceptions noted) are discussed in this section.

Subroutine Index

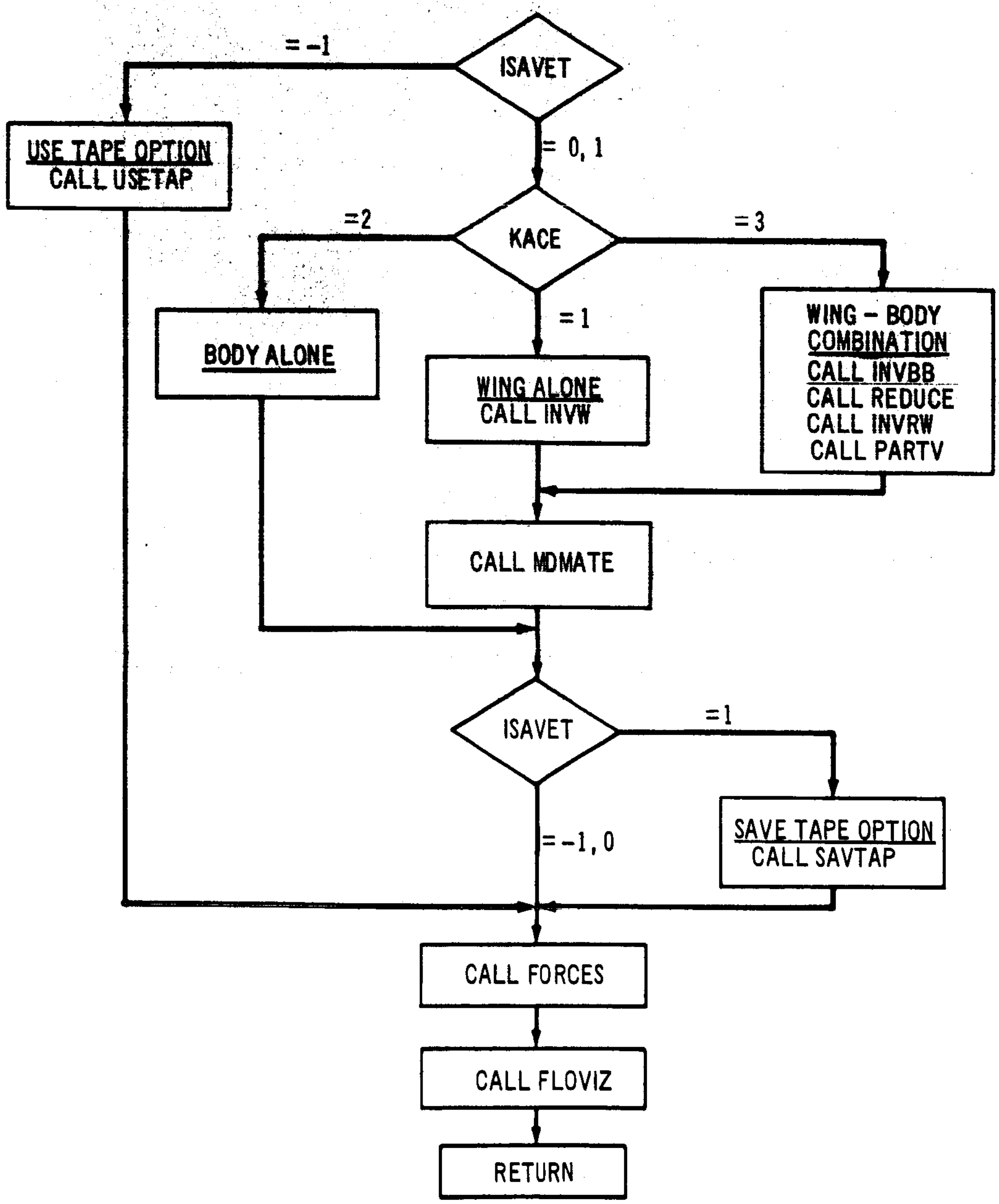
**SUBJECT:** **FORTRAN IV Subroutine ACOS**

**(See subroutine FASNCS)**

**SUBJECT:** FORTRAN IV Subroutine AERO

**PURPOSE:** To control the flow through the Aerodynamics section of the program.

**METHOD:** The flow to the main subroutines of the aerodynamics links is controlled by this subroutine for the three possible configurations (wing alone, body alone, and wing-body combinations).



USAGE: CALL AERO

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

Input: NTAPEC = Logical number of the tape on which geometrical data and aerodynamic matrices can be written and saved for later computer runs.

KACE = Code that indicates type of configuration.

= 1, wing alone.

= 2, body alone.

= 3, wing-body combination.

SUBPROGRAM CALLED:

INTURP
USETAP
AMATE
INVW
INVBB
REDUCE
INVRW
PARTV
MDMATE
SAVTAP
FORCES
FLOVIZ
UNLOAD

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE $240_8 = 360_{10}$

| | |
|---|---|
| **SUBJECT:** | **FORTRAN IV Subroutine ALLIN** |
| **PURPOSE:** | **To read a save tape to obtain necessary geometric and computational results from earlier parts of the program for use by EVAL1 and VEL1.** |
| **METHOD:** | **The tape is read according to the format given in Appendix D.** |

**USAGE:**

| | |
|---|---|
| CALL ALLIN | (NTA) |
| COMMON | /FLOV1/ KACE, NPANEL, NBODY, NWING, NBODYS, NWINGS, NROW, XMACH, SYM |
| COMMON | /FLOV2/ X, Y, Z, NPART, ALPHAS, THETA, XBB, R, WT, T, TC, SST, CHORD |
| COMMON | /FLOV3/ T 11, TC 11 |
| COMMON | /BA/ XB (210), YB (210), ZB (210) |
| COMMON | /BB/ XC (210), YC (210), ZC (210) |
| DIMENSION | X (210, 3, 4), Y (210, 3, 4), Z (210, 3, 4) |
| DIMENSION | NROW (2), NPART (210) |
| DIMENSION | ALPHAS (210), THETAS (210), CHORD (210) |
| DIMENSION | XBB (50), R (50), WT (120) |
| DIMENSION | T (50), TC (50), SST (210) |
| Inputs: | NTA = logical tape number of save tape |
| | XB = X-coordinate of boundary point |
| | YB = Y-coordinate of boundary point |
| | ZB = Z-coordinate of boundary point |
| | XC = X coordinate of centroid of panels |
| | YC = Y coordinate of centroid of panels |
| | ZC = Z coordinate of centroid of panels |

The arrays described above are not referenced and are included merely as a convenience for future modifications.

SUBPROGRAMS CALLED: FSF

ERROR RETURNS: None

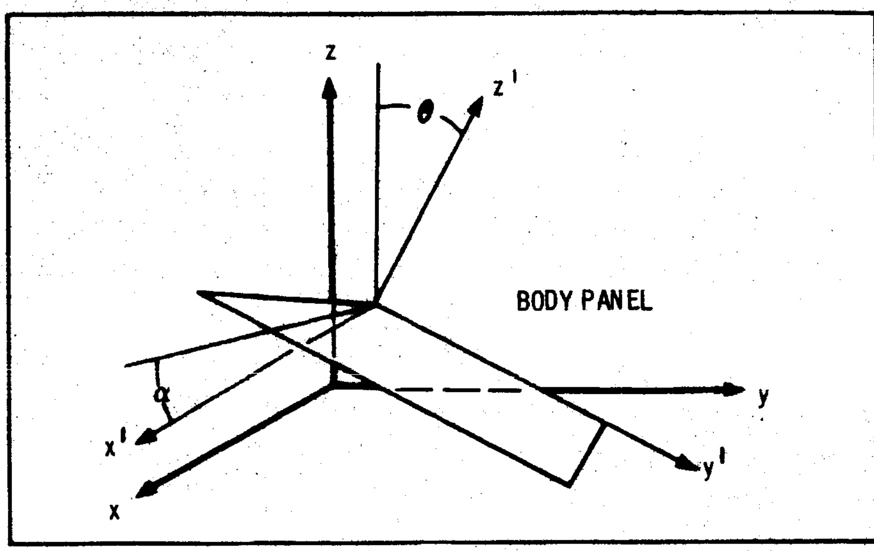
RESTRICTIONS: Assumes that the save tape is written in proper format.

STORAGE: $2200_8 = 1172_{10}$

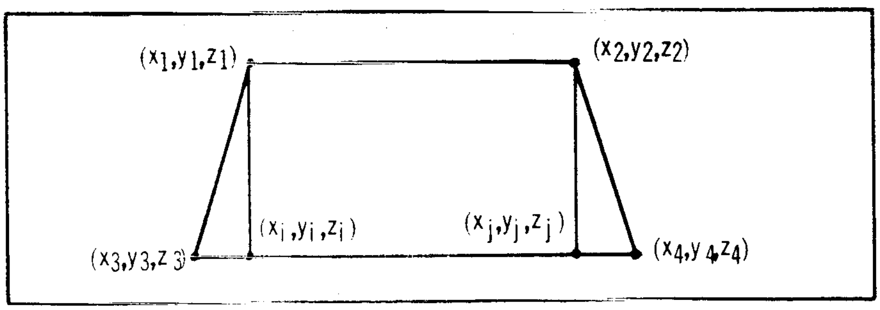
SUBJECT: FORTRAN IV Subroutine ALPHAB

PURPOSE: To calculate body panel alpha-incidence angles.

METHOD: The program calculates the panel alpha-incidence angle by use of a geometric function. For an arbitrary body panel,




the incidence angle, $\alpha$, is the angle of the panel in the x', y', z' coordinate system of the panel as viewed down the y'-axis. The x'-axis of this system is parallel to the x-axis of the body coordinate system, though this is not true for the y' and z' axes for a panel of nonzero $\theta$-inclination. For a body panel defined as follows,




the $\alpha$-incidence angle is calculated by the following formula, which involves the inclination angle $\theta$,

$$\alpha = \tan^{-1}\left[\frac{(z_1 - z_i)\cos\theta - (y_1 - y_i)\sin\theta}{(x_1 - x_i)}\right]$$

**USAGE:**

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
THETA

Output: ALPHA

CALL ALPHAB

**SUBPROGRAMS CALLED:**

| | |
|---|---|
| ATAN<br>SIN<br>COS | Built-in functions |

**ERROR RETURNS:** None

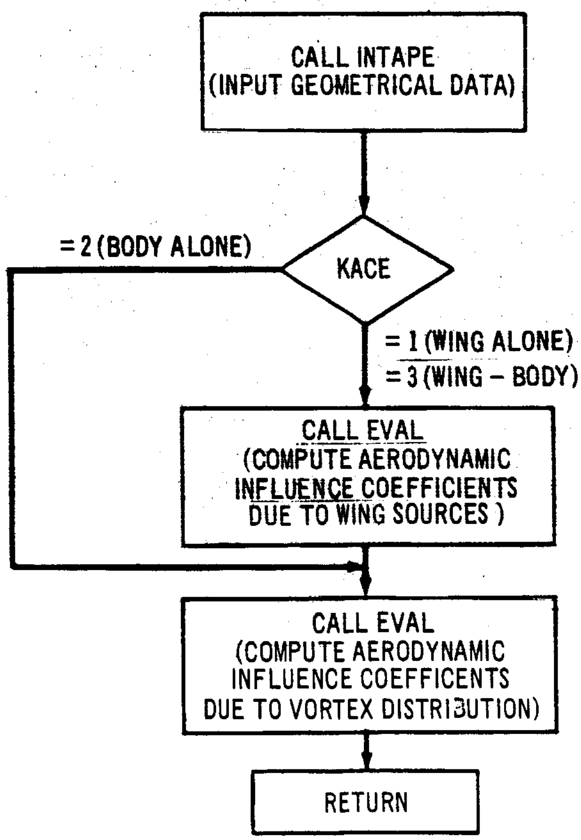
**STORAGE:** $362_{10} = 552_8$

SUBJECT: FORTRAN IV Subroutine AMATE

PURPOSE: To control the flow through a section of the aerodynamic links of the program.

METHOD: The flow is given in the following diagram.

SUBROUTINE AMATE



USAGE: CALL AMATE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

COMMON /BLOCK/ ALPHAS(210), AREA(210), A(210), ALPHAC(110), ALPHAT(110), CHORD(210), ISYM, NPART(210), NPANEL, NROW(2), THETA(210), TAIL, U(210), V(210), VPM(210), VV(210), VPMM(210), W(210), WPM(210), WW(210), WPMM(210), X(210,3,4), XBAR(210), XC(210), Y(210,3,4), YBAR(210), YC(210), Z(210,3,4), ZBAR(210), ZC(210)

Input:

| | | |
|---|---|---|
| KACE | = | Code that indicates type of configuration. |
| | = | 1, wing alone. |
| | = | 2, body alone. |
| | = | 3, wing-body combination. |
| NBODY | = | Number of body panels. |
| XC | = | Array of panel control point coordinates. |
| XBAR | = | Array of panel centroid coordinates. |

SUBPROGRAMS CALLED: INTAPE
EVAL

ERROR RETURNS: None

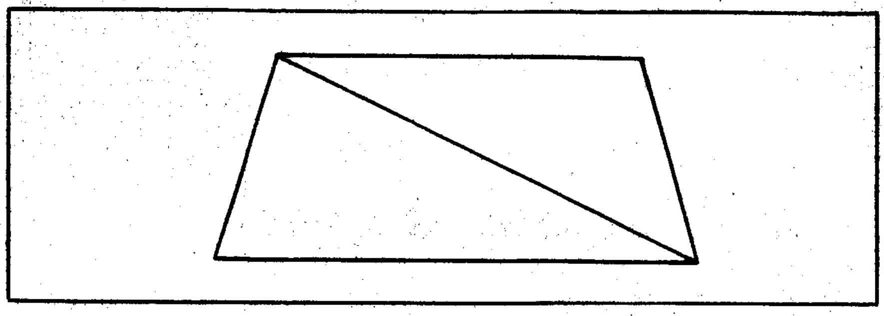
RESTRICTIONS: None

STORAGE: $12282_{10} = 27772_8$

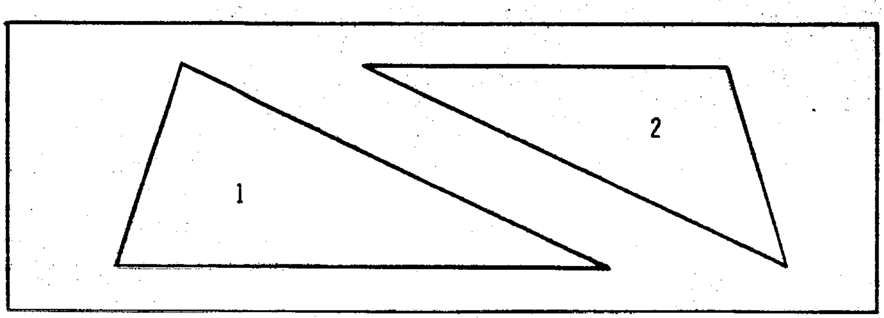
SUBJECT: FORTRAN IV Subroutine AREAP

PURPOSE: To calculate the area of a panel.

METHOD: Area S of a panel is obtained by dividing panel into two triangular subpanels, using a vector cross-product routine to calculate the area of each subpanel and summing these two results for the area of the panel. For example, a quadrilateral panel,



is divided into two subpanels, thus:




and areas $S_1$ and $S_2$ of each subpanel are calculated and summed for panel area S.

$$S_1 + S_2 = S$$

The subroutine also calculates ratio R of area $S_1$ of subpanel 1, to panel area S,

$$R = \frac{S_1}{S}$$

This value is used in other subroutines.

USAGE: DIMENSION X(16, 16), Y(16, 16), Z(16, 16), S(15, 15), R(15, 15)

$\left.\begin{matrix} X \\ Y \\ Z \end{matrix}\right\} =$ x, y, and z coordinates of panel corner points.

S = Panel area.

R = Ratio of subpanel area to panel area.

CALL AREAP (X, Y, Z, S, R, NC, NR)

Input: X, Y, Z (see above)
NC = Number of panel columns.
NR = Number of panel rows.

Output: S, R (see above)

SUBPROGRAM CALLED: VCROS

ERROR RETURNS: None

STORAGE: $239_{10} = 357_8$

SUBJECT: FORTRAN IV Subroutine ASIN

(See subroutine FASNCS)

| | |
|---|---|
| SUBJECT: | FORTRAN IV Function ATN1 |
| PURPOSE: | To compute the arctangent, in radians. |
| METHOD: | This function has been replaced by the standard built-in function ATAN2 with a $NAME control card. |

**SUBJECT:** FORTRAN IV Subroutine BCAM

**PURPOSE:** To compute camber slopes at body panel control points from a given body camber shape.

**METHOD:** The camber shape is specified by a table of z (camber) vs. x (body source control station). This table is converted to a second table of $\Delta z$ vs. x:

$$\Delta z_i = \frac{z_{i+1} - z_i}{x_{i+1} - x_i}$$

Slope $\alpha_c$ (camber slope) at xc (body panel control point) is contained by interpolation:

$$\alpha_{c_j} = \Delta z_{i+1} + \frac{xc_j - x_{i-1}}{x_i - x_{i-1}} \times (\Delta z_i - \Delta z_{i-1})$$

**USAGE:** CALL BCAM (NBODYS, NROWB, XB, ZDELTA, XC, DZDXB, ACB)

DIMENSION XB(55), ZDELTA(55), XC(100), DZDXB(55), ACB(55)

Input:

| | | |
|---|---|---|
| NBODYS | = | Number of body source control stations. |
| NROWB | = | Number of rows of body panels. |
| XB | = | Array of x-coordinates of body source control stations. |
| ZDELTA | = | Array of camber shapes at the body source control stations. |
| XC | = | Array of x-coordinates of body panel control points. |
| DZDXB | = | Dummy array. |

Output:

| | | |
|---|---|---|
| ACB | = | Array of camber slopes at the body panel control points. |

**SUBPROGRAMS CALLED:** None

**ERROR RETURNS:** None

**RESTRICTIONS:** NBODYS $\leq$ 55

**STORAGE:** $132_{10} = 204_8$

SUBJECT: FORTRAN IV Subroutine BCUTX

PURPOSE: To find the intersections of a set of body meridian lines with a plane normal to the x-axis.

METHOD: Subroutine POLXN is used to find the intersection of each meridian line with the plane.

USAGE: DIMENSION B(1), A(2, N)

CALL BCUTX (B, N, X, EP, NA, A)

Input: B = Array of body meridian line points, with header (see subroutine WBXUL).

N = Number of body meridian lines.

X = Body station. The plane will be taken as x = X.

EP = A tolerance used by POLXN in finding intersections.

Output: NA = Number of points in A. NA $\leq$ N.

A = Array of points ($y_i$, $z_i$) in section. These points are in 1-to-1 correspondence with meridian lines only if each meridian line intersects the plane.

SUBPROGRAM CALLED: POLXN

ERROR RETURNS: None

RESTRICTIONS: N must be greater than zero. If a meridian line has multiple plane intersections, the first intersection found will be used.

STORAGE: $133_{10} = 205_8$

**SUBJECT:** FORTRAN IV Subroutine BITURP

**PURPOSE:** Given a table of y versus x and a value of x, to find the corresponding value of y by either linear or biquadratic interpolation.

**METHOD:** Let n be the number of points $(x_i, y_i)$ in the given table. Let X be the given value of x for which the corresponding Y is desired. If n = 2, the linear interpolation mode is selected (even if biquadratic interpolation was requested). If $x_1 < x_n$, the table is searched from top to bottom; otherwise the search is from bottom to top. If X is outside the table range, an error code is set and Y is returned as $y_1$ or $y_n$ (depending on whether X is closer to $x_1$ or $x_n$). If $X = x_i$ then no interpolation is necessary and $Y = y_i$. Otherwise, the table interval that contains X is located. For linear interpolation, the calculation of Y is trivial.

Y may be found by biquadratic interpolation if X does not lie in the first or last table interval. For illustration of the method, assume that $x_2 < X < x_3$. Fit a quadratic $Q_2(x)$ to the points $(x_i, y_i)$, i = 1, 2, 3 and another quadratic $Q_3(x)$ to the points $(x_i, y_i)$, i = 2, 3, 4. Find the slopes $Q'_2(x_2)$ and $Q'_3(x_3)$. Fit a cubic to the points $(x_2, y_2)$ and $(x_3, y_3)$ with the corresponding slopes $Q'_2(x_2)$ $Q'_3(x_3)$. Evaluate the cubic at x = X to find Y.

However, Y is actually computed by the following interpolating formula (which has a continuous first derivative), mathematically equivalent to the above method:

$$Y = \frac{(x_3 - X)Q_2(X) + (X - x_2)Q_3(X)}{x_3 - x_2}$$

$Q_2(X)$ and $Q_3(X)$ are found by subroutine QRAT.

If X lies in the first or last interval of the table, biquadratic interpolation cannot be used. Y is therefore computed by quadratic interpolation.

USAGE:

DIMENSION X(1), Y(1), NU(3)

CALL BITURP (X, Y, I, N, K, XI, YI, NU)

Input: X = Location of the first x-coordinate in the table.

Y = Location of the first y-coordinate in the table.

I = Skip number; that is, the spacing of table values in the X and Y arrays. For example, I = 1 if X = $(x_1, x_2, \ldots)$ and Y = $(y_1, y_2, \ldots)$; however, if x and y are stored in a single array, say W = $(x_1, y_1, x_2, y_2, \ldots)$, then I = 2 and the subroutine call would be CALL BITURP (W, W(2), 2, ...).

N = Number of points in the table.

K = Interpolation code (K = 1 if linear interpolation; K = 2 if biquadratic interpolation).

XI = The given value of x for which the corresponding value of y is desired.

NU = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = NU(3) - 1. Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: YI = Interpolated value of y (if NU(1) = 0, 1, or 2).

NU = Error indicator array: NU(1) = 0 if success; = -1 if I $\leq$ 0; = -2 if N < 2; = -3 if $x_1 = x_n$; = 1 or 2 if X is outside the table range; = 3 or 4 if a quadratic could not be formed (because two points near the line x = X have the same x-coordinate. NU(2) is not an output. NU(3) is reduced by one if an error was detected.

SUBPROGRAM CALLED: QRAT

RESTRICTIONS: BITURP does not check the X array to see that it is monotonic. Weird results may be obtained if it is not monotonic in the neighborhood of the interpolating interval for biquadratic interpolation.

STORAGE: $375_{10} = 567_8$

**SUBJECT:** FORTRAN IV Subroutine BJTURP
(See Subroutine BITURP)

SUBJECT: FORTRAN IV Subroutine BLDM

PURPOSE: To compute the coefficients of lift, drag, and pitching moment for the region of the body represented by panels.

METHOD: The force normal to the panel is given as the product of the dynamic pressure, surface pressure coefficient, and panel area:

$$F_i = q \cdot C_{p_i} \cdot A_i$$

Resolving into the components normal and parallel to the free stream yields the lift and drag

$$L_i = - F_i \cdot \cos \theta_i$$

$$D_i = F_i \cdot n_i$$

where $n_i$ is the component of velocity normal to the x-axis and $\theta_i$ is the angle between the plane of the panel and a panel parallel to the x-y plane.

The moment of force with respect to a point (x, 0, z) is given by

$$M_i = - L_i (\bar{x}_i - x) + D_i (\bar{z}_i - z)$$

where $\bar{x}_i$, $\bar{z}_i$ are the x and z coordinates of the panel centroid.

Finally, the coefficients of lift, drag, and moment on the body panels are given by

$$C_L = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} L_i$$

$$C_D = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} D_i$$

$$C_M = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} M_i$$

where $S_W$ is the wing reference area, and NM is the number of body panels.

USAGE:

```
CALL BLDM(NM, XP, ZP, RFAREA, AREA, XBAR,
          ZBAR, ALPHAM, THETAM, CPM, CL,
          CD, CM)
```

| | | | |
|---|---|---|---|
| DIMENSION | AREA(NM), XBAR(NM), ZBAR(NM), ALPHAM(NM), THETAM(NM), CPM(NM) | | |
| Input: | NM | = | Number of body panels. |
| | XP | = | x-coordinate of the point about which the pitching moments are to be calculated. |
| | ZP | = | z-coordinate of the point about which the pitching moments are to be calculated. |
| | RFAREA | = | Wing reference area. |
| | AREA | = | Array of body panel areas. |
| | XBAR | = | Array of body panel centroid x-coordinates. |
| | ZBAR | = | Array of body panel centroid z-coordinates. |
| | ALPHAM | = | Array of the velocity components normal to the x-axis. |
| | THETAM | = | Array of angles between the plane of the panel and a plane parallel to the x-y plane. |
| | CPM | = | Array of panel surface pressure coefficients. |
| Output: | CL | = | Coefficient of lift. |
| | CD | = | Coefficient of drag. |
| | CM | = | Coefficient of pitching moment. |

SUBPROGRAM CALLED: COS (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $118_{10} = 166_8$

SUBJECT: FORTRAN IV Function BLUNDR

PURPOSE: This routine is used to prevent the overflow of arrays in GRIDS.

METHOD: Compares the product of increment parameters IX, IY, IZ against NMAX as indicated by MODE (see Subroutine GRIDS).

USAGE: IF (BLUNDR (IX, IY, IZ, MODE, NMAX).NE.O.) RETURN

Input: IX = x - increment parameters.

IY = y - increment parameters.

IZ = z - increment parameters.

MODE = Indicates the planar cut to be taken; the total number of points in such a plane must be less than NMAX.

NMAX = The maximum dimension of the GP and VLS arrays.

Output: BLUNDR = 0. acceptable inputs.

= 1. unacceptable inputs.

A message is written on the output tape if BLUNDR = 1

SUBROUTINES CALLED: None

ERROR RETURNS: See Output above.

RESTRICTIONS: None

STORAGE: $135_{10} = 217_8$

| | |
|---|---|
| **SUBJECT:** | FORTRAN IV Subroutine BODCR |
| **PURPOSE:** | To find the area, equivalent radius, and centroid of a body section. |
| **METHOD:** | It is assumed that the body is symmetrical about a plane parallel to the x-z plane, and that only one side of the body is given. The body is specified by a set of meridian lines (subroutine WBXUL). Subroutine BCUTX is used to find the y-z coordinates of the meridian lines at a given value of x. These coordinates form the two-dimensional points of a body section. A test is made to see that the y-coordinates of the first and last points are the same within a given tolerance. The average of these two values is used for the centroid y-coordinate. Subroutine ENRYCH is called and, if a given quantity CHD is greater than zero, additional points on the section are interpolated. The centroid and area of the section are found by subroutine CEGAR. The area is then doubled to make it apply to the entire body section. If the area is smaller than a given tolerance, the z-coordinate of the centroid is taken as the average of the z-coordinates of the first and last points in the section; otherwise, the value found by CEGAR is used. The z-coordinate of the centroid is set to zero if it is smaller than a given tolerance. The section radius is found by averaging the distances from the centroid to each meridian line; if the radius is less than a given tolerance, it is set to zero. |

USAGE:

DIMENSION B(1), EP(5), S(2, N), E(MAXE), NU(3)

CALL BODCR (B, N, X, EP, CHD, MAXE, S, E, NE, Y, Z, AREA, RAD, NU)

| | | |
|---|---|---|
| Input: | B | = Body meridian line array, with header (see subroutine WBXUL). |
| | N | = Number of meridian lines. |
| | X | = Body station (x-coordinate section). |
| | EP | = Array of tolerances. |
| | CHD | = Tolerance that regulates the number of points interpolated by ENRYCH (affects only AREA and Z). |
| | MAXE | = Length of array E. |

| | | | |
|---|---|---|---|
| | NU | = | Error indicator array. NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if $NU(2) \leq 0$. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if $NU(3) > 0$ and $NU(2) > 0$, an error message is written. |
| Output: | S | = | Array of points $(y_i, z_i)$ in section. |
| | E | = | Scratch array. |
| | NE | = | Number of elements in enriched array. |
| | Y | = | y-coordinate of centroid. |
| | Z | = | z-coordinate of centroid. |
| | AREA | = | Area of entire section (twice area of the half-section). |
| | RAD | = | Equivalent radius. |
| | NU | = | Error indicator array. NU(1) = 0 if success. |

SUBPROGRAMS CALLED

ENRYCH
CEGAR
SQRT (Built-in function)
BCUTX

ERROR RETURNS: A message is written if an error is detected. NU = 1 if the number of points in the section (found by BCUTX) is less than N or less than 2. An error of NU = 2, 4, or 5 is from ENRYCH. If an error code of 3 was returned by ENRYCH, NU is set to zero but a message is written.

RESTRICTIONS: CHD should be zero if the section is symmetrical about a horizontal plane or if there are less than eight meridian lines.

STORAGE: $486_{10} = 746_8$

**SUBJECT:** FORTRAN IV Subroutine BODY

**PURPOSE:** To serve as a control program for the body paneling subroutines.

**METHOD:** Program uses a series of FORTRAN IV language "CALL" statements to call the required body paneling subroutines.

The program contains the following labeled COMMON statement that occurs in all lower level body paneling subroutines, COMMON /COM2/ NPLNB, NPLNW, JLEAD, JTRAIL, IMID, NPTS(16), X(16, 90), Y(16, 90), Z(16, 90), XCEPT(21), XCEPTB(21), XCEPTW(16), YCEPTW(16), ZCEPTW(16), CODEBW(16), KPANEL(15, 20), XCOR(16, 21), YCOR(16, 21), ZCOR(16, 21), XINT(15, 20, 2), YINT(15, 20, 2), ZINT(15, 20, 2), XCEN(15, 20), YCEN(15, 20), ZCEN(15, 20), XCON(15, 20), YCON(15, 20), ZCON(15, 20), AREA(15, 20), ARAT(15, 20), THETA(15, 20), ALPHA(15, 20), CHORD(15, 20)

NPLNB = Number of cutting planes to intersect the body fore and aft of the body-wing intersection region.

NPLNW = Number of cutting planes to intersect the body in the body-wing intersection region.

JLEAD, JTRAIL, IMID = Codes for internal program control.

NPTS = Number of points on the successive body meridian lines that define the body.

X, Y, Z = x, y, and z coordinates of points on body meridian lines.

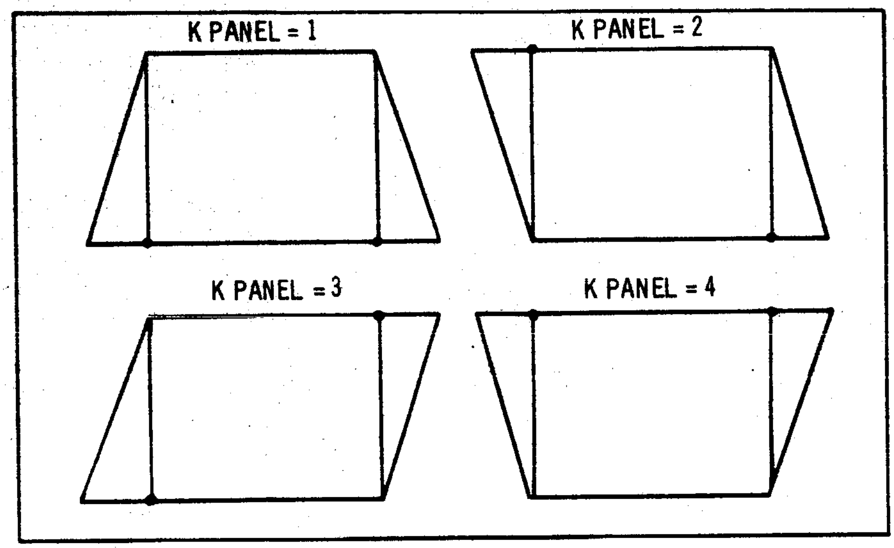
XCEPT = x intercepts of all body cutting planes.

XCEPTB = x intercepts of those body cutting planes that intersect fore and aft of the body-wing intersection region.

XCEPTW, YCEPTW, ZCEPTW = x, y, and z coordinates of the intersections with the body of those wing percent lines that define body cutting planes.

CODEBW = Codes that indicate wing percent lines that are to define body cutting planes in body-wing intersection region.

KPANEL = Code for body panel type. If KPANEL (I, J) = 1, both additional subpanel corner points are located on panel trailing edge. If KPANEL(I, J) = 2, inboard point is on leading edge, and outboard point is on trailing edge. If KPANEL (I, J) = 3, inboard point is on trailing edge and outboard point is on leading edge. If KPANEL(I, J) = 4, both points are on leading edge.




| | | |
|---|---|---|
| XCOR, YCOR, ZCOR | = | x, y, and z coordinates of body panel corner points. |
| XINT, YINT, ZINT | = | x, y, and z coordinates of additional subpanel corner points. |
| XCEN, YCEN, ZCEN | = | x, y, and z coordinates of body panel centroid. |
| XCON, YCON, ZCON | = | x, y, and z coordinates of body panel control point. |
| AREA | = | Body panel area. |
| ARAT | = | Ratio of subpanel area to panel area. |
| THETA | = | Body panel theta-inclination angle. |
| ALPHA | = | Body panel alpha-incidence angle. |
| CHORD | = | Body panel streamwise chord length. |

| | |
|---|---|
| USAGE: | COMMON (See subroutine OPCAMI for description of unlabeled COMMON) |
| | CALL BODY |
| SUBPROGRAMS CALLED: | INPUTB<br>CRNRB<br>AREAP<br>CENTRD<br>CNTRLB<br>THETAB<br>ALPHAB<br>OUTPTB |
| ERROR RETURNS: | Error message indicates whether error occurred in calculation of body panel geometry. |
| STORAGE: | $152_{10} = 230_8$ |

**SUBJECT:** FORTRAN IV Subroutine BODY1

**PURPOSE:** To read data cards that specify a body surface and generate three-dimensional points on body meridian lines.

**METHOD:** A body surface is specified by a set of defining sections normal to the X axis. Points on defining sections are given in polar coordinate form ($\rho$, $\theta$) or in rectangular coordinates (y, z). The number of points must be the same for all sections (except for zero-area sections). For convenience, several options are available for specifying section points (see Input Data Format, 3.3, card 6D).

Three-dimensional points on the $i^{th}$ body meridian line are formed by selecting the $i^{th}$ point at each defining section. Additional points are interpolated (if requested) so that straight-line connections between adjacent points will represent a smooth curve as closely as specified.

**USAGE:**

```
DIMENSION   DAT(2), B(1), AXIS(2), TITLE(12), NU(3)
LOGICAL   LGDEF(3, 6)
CALL BODY1 (DAT, LI, LO, B, LSTA, NSTA, LBPL,
            MBPL, NBPL, LTHETS, AXIS, TITLE,
            LGDEF, NU)
```

Input:

| | | |
|---|---|---|
| DAT | = | Date (alphameric). |
| LI | = | Input tape number. |
| LO | = | Output tape number. |
| B | = | Buffer for variable-length arrays. |
| NU | = | Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written. |

Output:

| | | |
|---|---|---|
| B | = | Buffer for variable-length arrays. |
| LSTA | = | The set of body-defining stations (x-values) starts in B(LSTA). |

| | | |
|---|---|---|
| NSTA | = | Number of body-defining stations. |
| LBPL | = | The body meridian line array, with header, starts in B(LBPL). See Appendix D for the array format. |
| MBPL | = | Number of meridian lines. |
| NBPL | = | Number of cells in meridian-line array. |
| LTHETS | = | The standard set of $\theta$ values (in degrees) corresponding to meridian lines, starts in B(LTHETS). |
| AXIS | = | y and z coordinates of the body main axis. |
| TITLE | = | Body title (alphameric). |
| LGDEF | = | LGDEF(1, 3) is set to .TRUE., LGDEF(2, 3) is set to .TRUE. if the body meridian lines are successfully computed. |
| NU | = | Error-indicator array. NU(1) is zero if no errors were detected. NU(3) may have been changed if an error was found (see INPUT, above). |

SUBPROGRAMS CALLED:

RICHNA
BODY1M
BODY1R
MERR
IDLETE
IRLEAS
IRSERV
IPACK

$\left.\begin{array}{l}\text{MERR}\\\text{IDLETE}\\\text{IRLEAS}\\\text{IRSERV}\\\text{IPACK}\end{array}\right\}$ (See Appendix A)

ERROR RETURNS: Function MERR is used to write an error message "ERROR i, CODE j IN SUBROUTINE BODY1 DURING GEOMETRIC DEFINITION" if an error is detected.

| i | j | Explanation |
|---|---|---|
| 1 | 0 | The number of defining sections (BNS, card 4D) or the number of meridian lines (BTHETA) is less than 2 or greater than 150. |

| i | j | Explanation (cont) |
|---|---|---|
| 2 | 0 | BNS * BTHETA is so large that the storage required for the meridian-line points (even with no enriching) is greater than the storage available in array B. |
| 3 | 1 | The code for the type of body section (SCODE, card 6D) is less than 0 or greater than 6. |
| 3 | 2 | SCODE = 3 (elliptical section), but a semi-axis is zero. |
| 4 | k | Meridian lines have been formed, but error k was detected in subroutine RICHNA. |
| 5 | 0 | Machine or program error of undetermined origin. |

RESTRICTIONS: The storage buffer, B, for variable-length arrays must have been initialized by subroutine INIBFR (Appendix A) or its equivalent.

STORAGE: $568_{10} = 1070_8$

SUBJECT: FORTRAN IV Subroutine BODY1M

PURPOSE: To write body meridian line points on an output tape.

METHOD: This is primarily an output routine for BODY1. The meridian lines are given in a single array with the format described in Appendix D. The polar coordinates of each point are computed in a plane normal to the x axis with polar origin at the body axis. Points that lie in a body defining section (within 0.0001) are identified by an asterisk.

USAGE: DIMENSION B(1), AXIS(2), TITLE(12), DAYT(2), STA(1)

CALL BODY1M (B, NB, AXIS, TITLE, DAYT, LO, STA)

Input:

| | | |
|---|---|---|
| B | = | Array of meridian lines, with header. |
| NB | = | Number of meridian lines. |
| AXIS | = | y, z coordinates of main body axis. |
| TITLE | = | Title (alphameric). |
| DAYT | = | Date (alphameric). |
| LO | = | Output tape number. |
| STA | = | Array of defining stations X-values). |

Output: Printout on tape LO.

SUBPROGRAMS CALLED:

ATN1
SQRT (Built-in function)
TRAV

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $373_{10} = 565_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine BODY1R |
| PURPOSE: | To read and write data that specify body defining sections, and form three-dimensional points on body meridian lines. |
| METHOD: | The number of body-defining sections, number of meridian lines, the Y and Z coordinates of the body axis, and a standard set of angles are given. Each section lies in a station plane (normal to the X axis). A meridian line of three-dimensional points on the body surface is formed by selecting corresponding points from each station. Let NPT = number of meridian lines = number of two-dimensional points per section = number of angles in the standard set.<br><br>Subroutine BODY1S is called for each section to read data cards and compute NPT three-dimensional points on the body surface. These points are then stored in meridian-line order. Results for each section are written on an output tape. |
| USAGE: | DIMENSION BAXIS(2), TITLE(12), DAT(2), THETS(NPT), THETA(NPT), RHO(NPT), PT(3, NPT), BPL(3, NSTA, NPT), STA(NSTA) |

CALL BODY1R (LI, LO, NPT, NSTA, BAXIS, TITLE, DAT, THETS, THETA, RHO, PT, BPL, STA, NU)

| | | |
|---|---|---|
| Input: | LI | = Input tape number. |
| | LO | = Output tape number. |
| | NPT | = Number of meridian lines. |
| | NSTA | = Number of defining sections (stations). |
| | BAXIS | = Y, Z coordinates of main body axis. |
| | TITLE | = Title (alphameric). |
| | DAT | = Date (alphameric). |
| | THETS | = Array of standard angles, in degrees. |
| Scratch: | THETA<br>RHO<br>PT | = Storage for $\theta$, $\rho$, and point coordinates (used at each section). |
| Output: | BPL | = Points on meridian lines. |
| | STA | = Array of defining stations (X coordinates). |

NU = Error indicator, which is zero for success.

SUBPROGRAM CALLED: BODY1S

ERROR RETURNS: If NU $\neq$ 0, see BODY1S.

RESTRICTIONS: Tape LI must be properly positioned and contain NSTA sets of data for BODY1S.

STORAGE: $368_{10} = 560_8$

**SUBJECT:** FORTRAN IV Subroutine BODY1S

**PURPOSE:** To read body-defining section data cards and compute points on the body section.

**METHOD:** Data cards are described in section 3.3. The code for section type is checked for validity; then the indicated operation is carried out.

**USAGE:**

DIMENSION BAXIS(2), THETS(NPT), SECAX(2), THETA(NPT), RHO(NPT), PT(3, NPT)

CALL BODY1S (LI, BAXIS, NPT, THETS, SECAX, SCODE, THETA, RHO, PT, NU)

Input:

- LI = Input tape number.
- BAXIS = Y, Z coordinates of the main body axis.
- NPT = Number of points in section.
- THETS = Array of angles, in degrees, which are used for SCODE = 1, 2, and 3.
- PT = See OUTPUT, below; used as input only if SCODE = 0.

Output:

- SECAX = Y, Z coordinates of the section origin ($\rho$, $\theta$ origin).
- SCODE = Code for type of section (read from card).

  0. This section is identical to the previous section, except for X-coordinates. Store the station value (read from card) as the X-coordinate for each point in PT.

  1. Read NPT values of $\rho$ from cards. Use with the THETS array to compute Y, Z coordinates of points on the section.

  2. Circular section. Use the radius (read from card) as $\rho$, and THETS to compute section points.

  3. Section is a rectangular ellipse. Use the semi-axes (read from card) to construct ellipse. Find

$\rho$'s corresponding to THETS. Construct points on section.

4. Circular section, but read NPT values of $\theta$ from cards instead of using THETS.

5. Read NPT values of $\rho$ from cards, then read NPT values of $\theta$ from cards.

6. The y,z coordinates of the section points are given. Find corresponding $\rho$, $\theta$ coordinates; but if any $\rho$ is zero, use corresponding $\theta$ from THETS.

THETA = Array of $\theta$ values (in degrees) corresponding to each section point (same as THETS for SCODE = 1, 2, 3).

RHO = Array of $\rho$ values.

PT = Array of points (x, y, z coordinates) in section.

NU = Error indicator, which is zero for success.

SUBPROGRAMS CALLED:

ATN1
UVECN
ELLIPR
SIN (Built-in function)
TRAV
COS (Built-in function)

ERROR RETURNS: NU = 1 if SCODE $<$ 0 or SCODE $>$ 6.
NU = 2 if SCODE = 3 and a semi-axis is zero.

RESTRICTIONS: Input tape LI is assumed to be correctly positioned and to contain valid data. Angles are measured from the vertical, and it is assumed that $0 \leq \theta \leq 180$. The $\rho$-$\theta$ origin is at SECAX.

STORAGE: $439_{10} = 667_8$

SUBJECT: FORTRAN IV Subroutine BSCALE

PURPOSE: To scale body meridian lines so that each line extends from $x = 0$ to $x = x_E$, where $x_E$ is given. The y and z coordinates are unchanged.

METHOD: The x-coordinates of the points on each meridian line are scaled as follows:

$$x' = (x - x_1)\, x_E/(x_L - x_1)$$

where $x_1$ = x of first point, $x_L$ = x of last point.

USAGE:

DIMENSION B(1)

CALL BSCALE (B, NB, XEND, NU)

| | | |
|---|---|---|
| Input: | B | = Meridian line array (see subroutine TFLAT1). |
| | NB | = Number of meridian lines. |
| | XEND | = $x_E$. |
| Output: | B | = Scaled meridian line array. |
| | NU | = Error indicator. NU = 0 if success. NU = k if $x_L - x_1 = 0$ for the $k^{th}$ meridian line. |

SUBPROGRAM CALLED: None

ERROR RETURNS See Output

RESTRICTIONS: None

STORAGE: $125_{10} = 175_8$

**SUBJECT:** FORTRAN IV Subroutine BTHICK

**PURPOSE:** To compute thickness slopes at body panel control points from a given body radii distribution.

**METHOD:** The radii distribution is specified by a table of r (radius) vs. x (body source control station). This table is used to obtain a second table of $\Delta r$ vs. x:

$$\Delta r_i = \frac{r_{i+1} - r_i}{x_{i+1} - x_i}$$

from which $\alpha_T$ (thickness slope) at xc (body panel control point) are calculated by interpolation:

$$\alpha_{T_j} = \Delta r_{i-1} + \left(\frac{xc_j - x_{i-1}}{x_i - x_{i-1}}\right) \cdot (\Delta r_i - \Delta r_{i-1})$$

**USAGE:** CALL BTHICK (NBODYS, NBODY, NROWB, XB, R, XC, ALPHA, THETA)

DIMENSION XB(55), R(55), XC(100), ALPHA (55), THETA(55)

Input:

- NBODYS = Number of body source control stations.
- NBODY = Number of body panels.
- NROWB = Number of rows of body panels.
- XB = Array of x-coordinates of body source control stations.
- R = Array of radii at the body source control stations.
- XC = Array of x-coordinates of body panel control points.
- ALPHA = Dummy array.

Output:

- THETA = Array of thickness slopes at the body panel control points.

**SUBPROGRAMS CALLED:** None

**ERROR RETURNS:** None

| | |
|---|---|
| **RESTRICTIONS:** | **NBODYS $\leqq$ 55** |
| | **NBODY $\leqq$ 100** |
| **STORAGE:** | **$180_{10} = 264_8$** |

SUBJECT: FORTRAN IV Subroutine CAMBER

PURPOSE: To compute the wing surface shape, given the wing surface slopes.

METHOD: The wing surface shape is computed column-by-column from the inboard column of wing panels to the outboard column. For a given column of wing panels, the surface shape is obtained panel-by-panel from the wing leading edge to the trailing edge. If i specifies panel row and j the column, then the surface shape is calculated:

$$s_{i,j} = \left\{ \sum_{k=1}^{j} \alpha_{w_{i,k}} \cdot c_{i,k} \right\} \Big/ C_i$$

where

| | | |
|---|---|---|
| $s_{i,j}$ | = | panel surface shape. |
| $\alpha_{w_{i,j}}$ | = | panel surface slope. |
| $c_{i,j}$ | = | panel chord length. |
| $C_i$ | = | panel column chord length. |

USAGE: CALL CAMBER (NCOLW, NROWW, CHORD, ALPHAW, CSHAPE, CHORDL)

DIMENSION: CHORD(110), ALPHAW(110), CSHAPE(110), CHORDL(110)

| | | | |
|---|---|---|---|
| Input: | NCOLW | = | Number of columns of wing panels. |
| | NROWW | = | Number of rows of wing panels. |
| | CHORD | = | Array of wing panel chord lengths. |
| | ALPHAW | = | Array of wing surface slopes. |
| | CHORDL | = | Dummy array. |
| Output: | CSHAPE | = | Array of wing surface shapes. |

SUBPROGRAMS CALLED: None

ERROR RETURNS: None

| | |
|---|---|
| **RESTRICTIONS:** | $(NCOLW \cdot NROWW) \leq 110$ |
| **STORAGE:** | $139_{10} = 213_8$ |

**SUBJECT:** FORTRAN IV Subroutine CAMBW

**PURPOSE:** To compute the velocity components normal to the x-axis on the wing panels, given the matrix of the aerodynamic influence coefficients and the pressure difference across the panels.

**METHOD:** The velocity components are given by

$$n_{W_i} = \sum_{i=1}^{NW} \sum_{j=1}^{NW} A_{WW_{ij}} \cdot p_{W_j}$$

where $A_{WW}$ is the matrix of aerodynamic influence coefficients (influence on the wing, due to the wing) and $p_W$ is the pressure difference across each wing panel.

**USAGE:** CALL CAMBW (NW, NT, A, CL, ALPHAW)

DIMENSION A(NW), CL(NW), ALPHAW(NW)

| | | |
|---|---|---|
| Input: | NW | = Number of wing panels. |
| | NT | = Logical tape number on which $A_{WW}$ is stored. |
| | A | = Dummy array used by the subroutine. |
| | CL | = Array of pressure differences. |
| Output: | ALPHAW | = Array of velocity components. |

**SUBPROGRAM CALLED:** None

**ERROR RETURNS:** None

**RESTRICTIONS:** None

**STORAGE:** $90_{10} = 132_8$

**SUBJECT:** FORTRAN IV Subroutine CAMBWB

**PURPOSE:** In the presence of a body, to compute the velocity components on the wing, normal to the x-axis.

**METHOD:** The velocity component on the wing panel i is given by

$$n_{W_i} = \sum_{j=1}^{NW} A_{R_{ij}} \cdot p_{W_j} + \sum_{j=1}^{NB} D_{ij} \cdot n_{B_j}$$

where $A_R$ = The matrix of the reduced aerodynamic influence coefficients,

$p_W$ = The pressure difference across wing panels,

D = The matrix formed by the product.

$$A_{WB} \cdot A_{BB}^{-1}$$

where $A_{WB}$ = A partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the wing.

$A_{BB}$ = The partition resulting from the influence of the body on the body.

$n_B$ = The velocity component on the body normal to the x-axis.

**USAGE:**

```
DIMENSION  A(NW), CLW(NW), ALPHAB(NB), ALPHAW(NW)
CALL  CAMBWB (NB, NW, NTAPEX, NTAPEY, A, CLW,
ALPHAB, ALPHAW)
```

Input:

NB = Number of body panels.

NW = Number of wing panels.

NTAPEX = Logical tape number on which $A_R$ is stored.

NTAPEY = Logical tape number on which D is stored.

A = Dummy array used by the subroutine.

CLW = Array of the pressure differences across wing panels.

ALPHAB = Array of velocity components on the body, normal to the x-axis.

Output: ALPHAW = Array of velocity components on the wing, normal to the x-axis.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $144_{10} = 220_8$

SUBJECT: FORTRAN IV Subroutine CEGAR

PURPOSE: To find the area and the centroid (center of gravity) of a polygon.

METHOD: Given n points ($x_i$, $y_i$: $i = 1, \ldots, n$) that are the vertices of a polygon, the area of the polygon is given by the magnitude of A, where

$$A = 1/2 \sum_{i=1}^{n} (y_{i+1} + y_i)(x_{i+1} - x_i)$$

where $x_{n+1}$, $y_{n+1}$ are set equal to $x_1$, $y_1$ by the subroutine to ensure that the polygon is closed. If any two or more of the points ($x_i$, $y_i$) coincide, no difficulty will result in the calculation of the centroid, and the area as long as the resulting polygon is a simple closed curve.

The coordinates of the centroid are given by,

$$\bar{x} = \sum_{i=1}^{n} (x_{i+1}^2 + x_i^2 + x_{i+1} x_i)(y_{i+1} - y_i) / (-6A), \text{ and}$$

$$\bar{y} = \sum_{i=1}^{n} (y_{i+1}^2 + y_i^2 + y_{i+1} y_i)(x_{i+1} - x_i) / (6A)$$

USAGE: DIMENSION P(2, n+1), CG(2)

CALL CEGAR (N,P,A,CG)

Input: N = the number of points, n.

P = the array of points ($x_i$, $y_i$) that are the vertices of the polygon.

Output: A = the magnitude of the area of the polygon.

CG = the centroid of the area, $(\bar{x}, \bar{y})$.

RESTRICTIONS: The polygon must be a simple closed curve, (i.e., the polygonal arc cannot cross itself).

ERROR RETURNS: If the area of the polygon is zero, then the subroutine sets the coordinates of the centroid equal to zero.

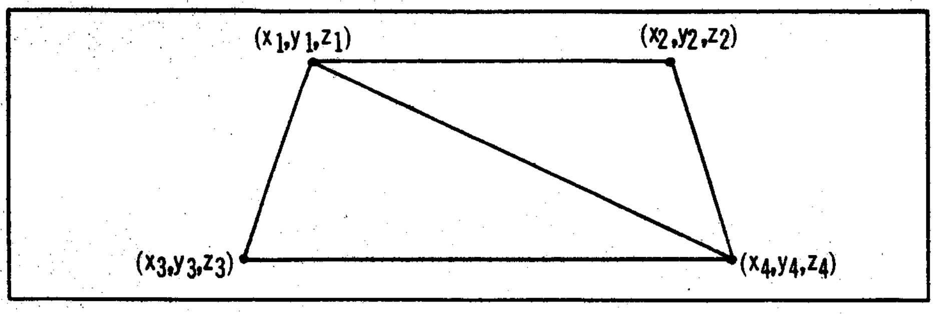
Because, in the calculation of $\bar{x}$ and $\bar{y}$, A is used as a divisor, considerable error could result in the values obtained for $\bar{x}$ and $\bar{y}$ when A is close to zero. Overflow could also occur for small values of A.

STORAGE: $229_{10} = 345_8$

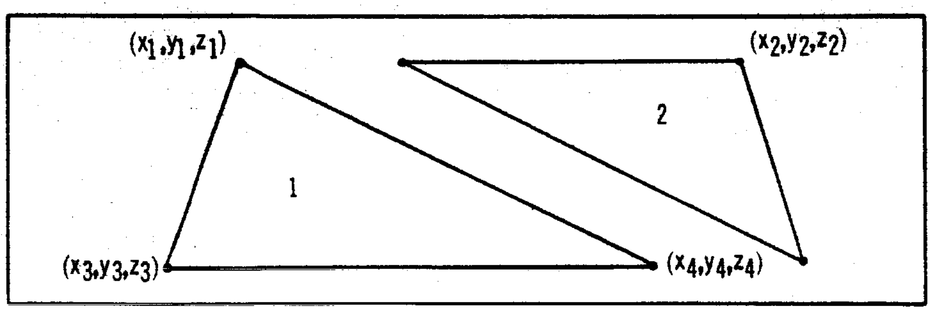
SUBJECT: FORTRAN IV Subroutine CENTRD

PURPOSE: To calculate coordinates of a panel centroid.

METHOD: Centroid coordinates of a panel are calculated by dividing panel into two subpanels, locating a centroid on each subpanel, and averaging coordinates of the two centroids with appropriate weighting factor. For example, a given quadrilateral panel:




is subdivided into two subpanels, thus:



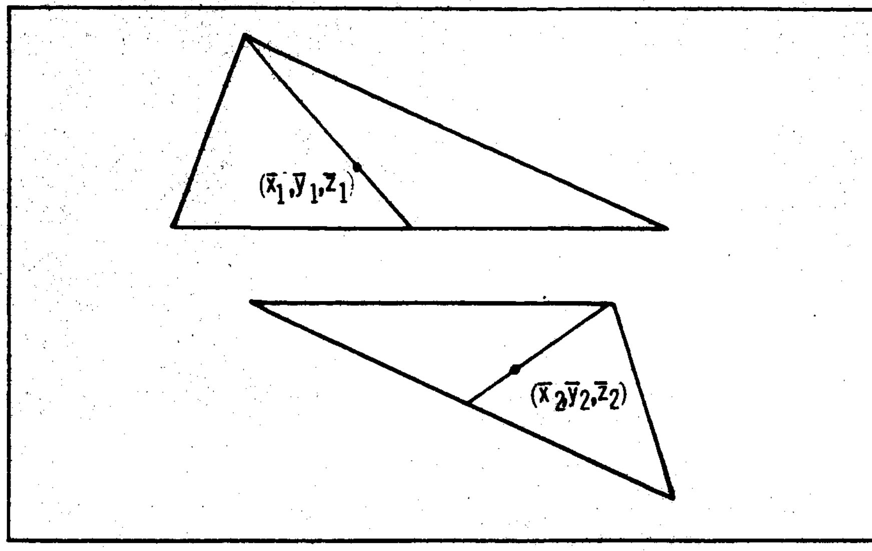

and a centroid is calculated for each subpanel using standard formulae. For subpanel 1 of the above example,

$$\bar{x}_1 = x_1 + \frac{2}{3}\left(x_3 + \frac{1}{2}(x_4 - x_3) - x_1\right)$$
$$\bar{y}_1 = y_1 + \frac{2}{3}\left(y_3 + \frac{1}{2}(y_4 - y_3) - y_1\right)$$
$$\bar{z}_1 = z_1 + \frac{2}{3}\left(z_3 + \frac{1}{2}(z_4 - z_3) - z_1\right)$$

Similarly for subpanel 2,

$$\bar{x}_2 = x_2 + \frac{2}{3}\left(x_1 + \frac{1}{2}(x_4 - x_1) - x_2\right)$$
$$\bar{y}_2 = y_2 + \frac{2}{3}\left(y_1 + \frac{1}{2}(y_4 - y_1) - y_2\right)$$
$$\bar{z}_2 = z_2 + \frac{2}{3}\left(z_1 + \frac{1}{2}(z_4 - z_1) - z_2\right)$$

This is shown in the following sketch,




A weighting factor is used to average coordinates of centroids of the two subpanels and derive coordinates of panel centroid. The weighting factor is merely the ratio of the area of sub-panel 1 to the panel area (see subroutine AREAP). That is,

$$\bar{x} = \bar{x}_2 + R(\bar{x}_1 - \bar{x}_2)$$

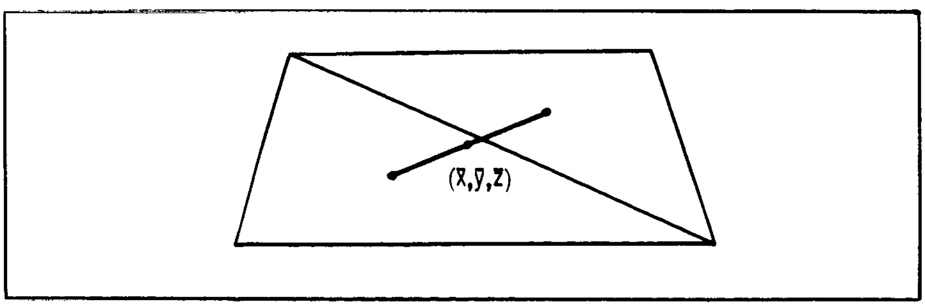
$$\bar{y} = \bar{y}_2 + R(\bar{y}_1 - \bar{y}_2)$$

$$\bar{z} = \bar{z}_2 + R(\bar{z}_1 - \bar{z}_2)$$

and,

$$R = \frac{S_1}{S}$$

This is shown below,




USAGE: DIMENSION X(16, 16), Y(16, 16), Z(16, 16), R(15, 15), XC(15, 15), YC(15, 15), ZC(15, 15)

X, Y, Z = x, y, and z-coordinates of panel corner points.

R = Ratio of subpanel area to panel area.

XC, YC, ZC =x-, y-, and z-coordinates of panel centroid.

CALL CENTRD (X, Y, Z, R, XC, YC, ZC, NC, NR)

Input: X, Y, Z, R (see above)

NC = Number of panel columns.
NR = Number of panel rows.

Output: XC, YC, ZC (see above)

SUBPROGRAM CALLED: None

ERROR RETURNS: None

STORAGE: $298_{10} = 452_8$

**SUBJECT:** FORTRAN IV Subroutine CHECK

**PURPOSE:** To check if a sequence $\{x_i\}$ of numbers is monotonic increasing (or decreasing).

**METHOD:** The given sequence $\{x_i\}$ can be one of four types:

1) Monotonic increasing; i.e., $x_i \leq x_{i+1}$
2) Strictly monotonic increasing; i.e., $x_i < x_{i+1}$
3) Monotonic decreasing; i.e., $x_i \geq x_{i+1}$
4) Strictly monotonic decreasing; i.e., $x_i > x_{i+1}$

The program calling sequence includes a code that indicates which type the sequence should be. The program consists of a "do-loop" involving an "if" statement that checks the sequence accordingly.

USAGE: CALL CHECK (X, N, K, L, N6)

Input:
- X = Sequence to be checked.
- N = Number of elements in sequence $\{x_i\}$.
- K = Code indicating sequence type that $\{x_i\}$ should be. If K = 1, $\{x_i\}$ should be monotonic increasing sequence. If K = 2, $\{x_i\}$ should be strictly monotonic increasing sequence. If K = -1, $\{x_i\}$ should be monotonic decreasing sequence. If K = -2, $\{x_i\}$ should be strictly monotonic decreasing sequence.
- N6 = Output tape unit.

Output:
- L = Error code. If L = 0, $\{x_i\}$ successfully checked. If L = 1, error occurred.

SUBPROGRAM CALLED: None

ERROR RETURNS: See above

**RESTRICTIONS:** The sequence $\{x_i\}$ must be a sequence of floating point numbers.

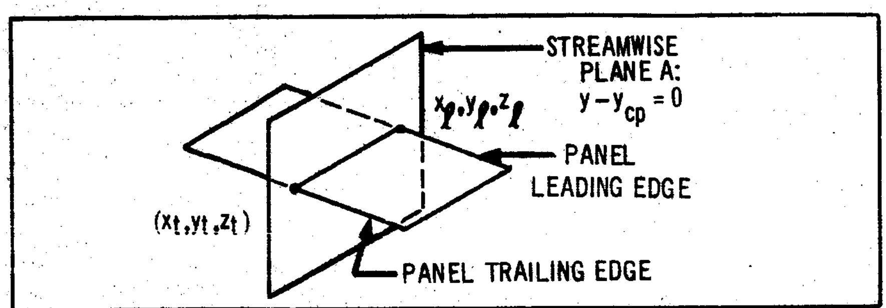
**STORAGE:** $237_{10} = 355_8$

SUBJECT: FORTRAN IV Subroutine CHORDW

PURPOSE: To calculate the length of a streamwise chord through the wing panel control point.

METHOD: The program calculates the length of the streamwise chord by use of a basic distance formula.

The chord is located by construction of a streamwise plane (A) normal to the plane of the panel and passing through the panel control point. This is shown in the following sketch:




The line of intersection of this plane and the original panel is the desired chord, and the chord length is calculated,

$$L = x_t - x_\ell$$

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: XCOR
YCOR
ZCOR
YCON

Output: CHORD

CALL CHORDW

SUBPROGRAM CALLED: POLXN

ERROR RETURNS: Error messages indicate if error occurred in use of subroutine POLXN to calculate chord intercepts.

STORAGE: $418_{10} = 642_8$

SUBJECT: FORTRAN IV Subroutine CLOK

PURPOSE: To find the time of day.

METHOD: This is a dummy subroutine which returns blanks. An appropriate change may be made at the user's installation to return the true time and date.

USAGE: CALL CLOK (TIME, DATE)

Output: TIME = Time of day (alphameric).

DATE = Date (alphameric).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

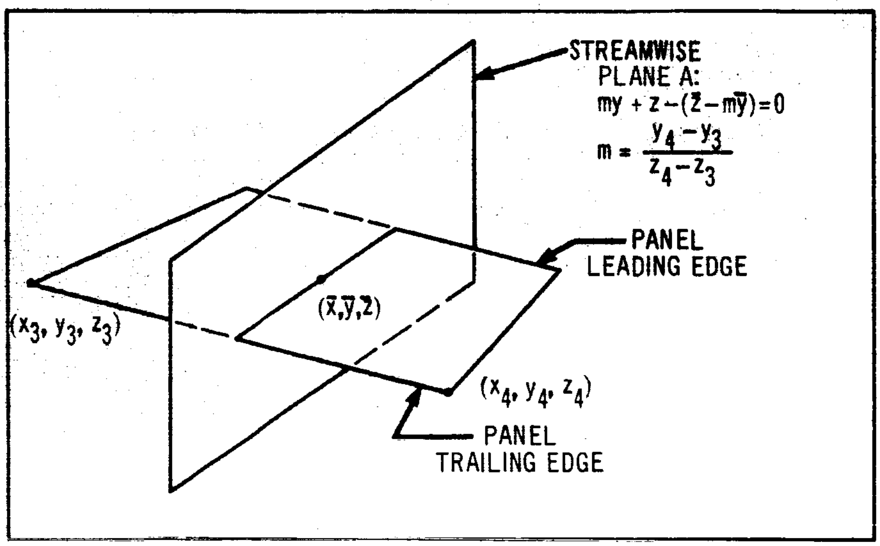
STORAGE: $23_{10} = 27_8$

**SUBJECT:** FORTRAN IV Subroutine CNTRLB

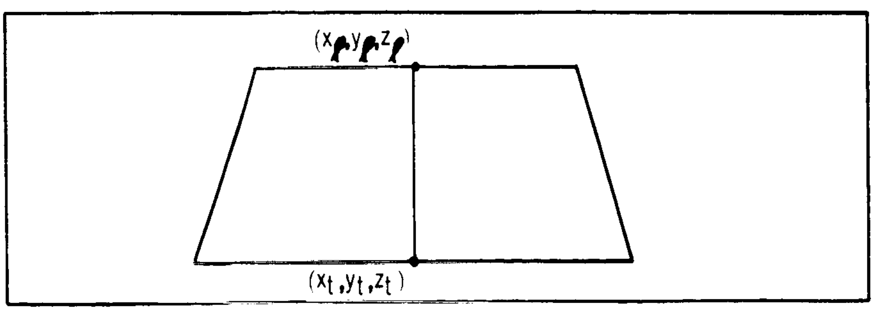
**PURPOSE:** To calculate the coordinates of a body panel control point and the length of a streamwise chord through the control point.

**METHOD:** The coordinates of a body panel control point are calculated by locating a streamwise chord through the panel centroid, determining the intercepts of this chord with the panel leading and trailing edges, and using a specified input constant to average these coordinates and locate the control point along the chord.

To find the chord that is required, the program constructs a streamwise plane (A) normal to the plane of the panel and through the panel centroid. For example,



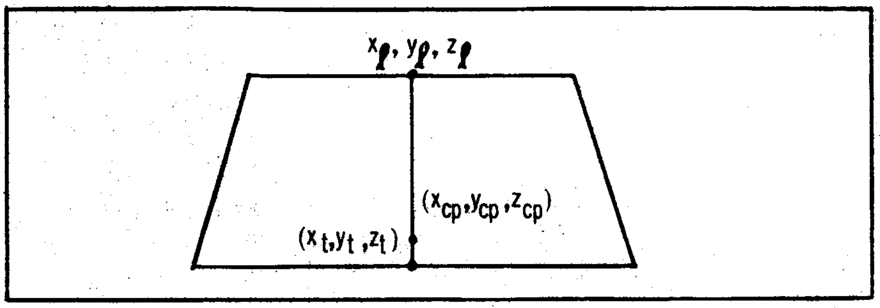

The intersection of this plane with the given panel defines the chord that is needed, and the chord length is calculated using a standard distance formula. For the above example,



and,

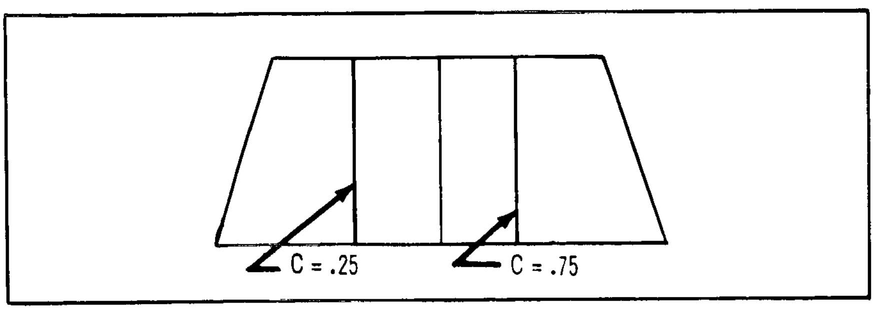
$$L = \left[(x_t - x_\ell)^2 + (y_t - y_\ell)^2 + (z_t - z_\ell)^2\right]^{1/2}$$

The control point coordinates are then calculated by using an input constant, c, and averaging the chord intercepts.




$$x_{cp} = x_\ell + c \cdot (x_t - x_\ell)$$
$$y_{cp} = y_\ell + c \cdot (y_t - y_\ell)$$
$$z_{cp} = z_\ell + c \cdot (z_t - z_\ell)$$

As noted above, the location of the panel control point along the streamwise chord is controlled by an input constant, c, (fraction). In addition and as an option, a second constant can be employed as input to change the spanwise location of the chord on which the control point is calculated. For a case in which the option is not used, the streamwise chord is constructed through the panel centroid; thus is located at approximately 50 percent in the spanwise direction. But for a case in which this option is used, this spanwise percent value can be changed to an arbitrary one, say 25 or 75 percent.



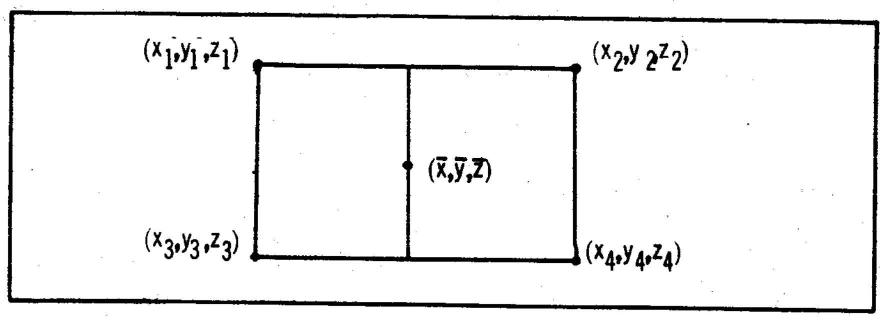
| | |
|---|---|
| USAGE: | COMMON (See subroutine OPCAMI for unlabeled COMMON description). |
| | CONTROL /COM1/ (See subroutine PANEL) |
| | Input: KODEC<br>XPER<br>YPER<br>NPER1<br>NPLN1 |
| | Output: KODEB |
| | COMMON /COM2/ (See subroutine BODY) |
| | Input: XCOR<br>YCOR<br>ZCOR<br>YCEN<br>ZCEN |
| | Output: XCON<br>YCON<br>ZCON<br>CHORD |
| | CALL CNTRLB |
| SUBPROGRAMS CALLED: | POLXN<br>SQRT (Built-in function) |
| ERROR RETURNS: | Error message indicates if error has been encountered in use of subroutine POLXN to calculate chord intercepts. |
| STORAGE: | $484_{10} \stackrel{>}{=} 744_8$ |

**SUBJECT:** FORTRAN IV Subroutine CNTRLW

**PURPOSE:** To calculate the coordinates of a wing panel control point.

**METHOD:** The coordinates of a wing panel control point are calculated by defining a constant, $c_1$, that locates the control point in the spanwise direction, and a second constant, $c_2$, that locates it in the streamwise direction. The latter constant, $c_2$, is always given as input data, though the former constant, $c_1$, can be given as input or calculated by the program as a function of the centroid location.

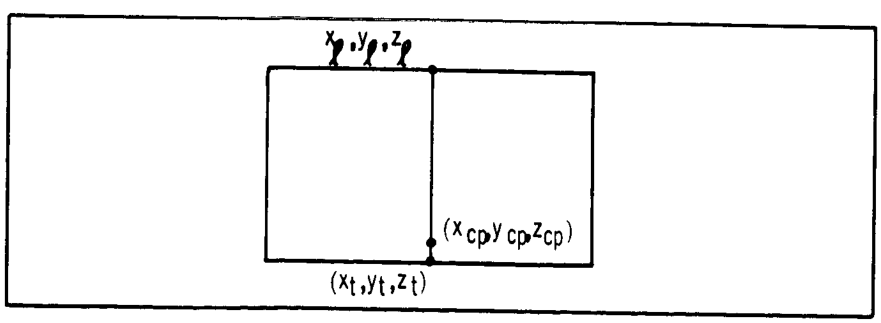
The two constants are used to average the panel corner point coordinates and derive the control point coordinates. As an example, a wing panel is defined,




The constant, $c_2$, is given as input and the constant, $c_1$, is either given as input or calculated by the program. For the latter case,

$$c_1 = \frac{\bar{y} - y_1}{y_2 - y_1}$$

These constants are then used to average the four corner point coordinates and obtain the control point coordinates,



$$x_{cp} = x_\ell + c_2 \cdot (x_t - x_\ell)$$
$$x_\ell = x_1 + c_1 \cdot (x_2 - x_1)$$
$$x_t = x_3 + c_1 \cdot (x_4 - x_3)$$
$$y_{cp} = y_\ell + c_2 \cdot (y_t - y_\ell)$$
$$y_\ell = y_1 + c_1 \cdot (y_2 - y_1)$$
$$y_t = y_3 + c_1 \cdot (y_4 - y_3)$$
$$z_{cp} = z_\ell + c_2 \cdot (z_t - z_\ell)$$
$$z_\ell = z_1 + c_1 \cdot (z_2 - z_1)$$
$$z_t = z_3 + c_1 \cdot (z_4 - z_3)$$

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: KODEC
XPER
YPER
KSTART
KEND
NPER1
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: XCOR
YCOR
ZCOR
YCEN

Output: XCON
YCON
ZCON

CALL CNTRLW

SUBPROGRAM CALLED: AMAX (Built-in function)

ERROR RETURNS: Error message indicates if error occurred in calculation of panel control point.

STORAGE: $735_{10} = 1337_8$

**SUBJECT:** FORTRAN IV Subroutine COEFS

**PURPOSE:** To compute the axial force coefficient, normal force coefficient and pitching moment coefficient on a body.

**METHOD:** For the axial force coefficient,

$$C_A = \frac{1}{S}\int_0^{\ell} r \cdot \frac{dr}{dx}\left[\int_0^{2\pi} C_p \, d\theta\right]dx;$$

for the normal force coefficient,

$$C_N = \frac{1}{S}\int_0^{\ell} r \left[\int_0^{2\pi} C_p \cos\theta \, d\theta\right]dx$$

and for the pitching moment coefficient,

$$C_M = \frac{1}{S\bar{c}}\int_0^{\ell} x \cdot r\left[\int_0^{2\pi} C_p \cos\theta \, d\theta\right]dx$$

where $x$ = x-coordinate of body station.

$r$ = Body radius at body station.

$\frac{dr}{dx}$ = Body surface first derivative at body station.

$C_p$ = Pressure coefficient at a body station and on a body meridian line.

$\Theta$ = Angle of body meridian line.

$\ell$ = Body length.

$S$ = Reference area.

$\bar{c}$ = Reference chord length.

For each coefficient, subroutine SIMUN3 is first used to obtain a value for the integral around the body (with respect to $\theta$) at each body station and then for the integral along the body axis (with respect to $x$ ). This result is multiplied by the appropriate constant to obtain the final value.

USAGE: For the axial force coefficient:

CALL SCX (XB, R, FD, THETAB, CPBB, NFT, NLT, NTHETA, XP, ZP, C, IRROR)

For the normal force coefficient:

CALL SCN (XB, R, FD, THETAB, CPBB, NFT, NLT, NTHETA, XP, ZP, C, IRROR)

For the pitching moment coefficient:

CALL SCM (XB, R, FD, THETAB, CPBB, NFT, NLT, NTHETA, XP, ZP, C, IRROR)

DIMENSION XB(55), R(55), FD(55), THETAB(10), CPBB(10, 55)

Input:

| | | |
|---|---|---|
| XB | = | Array of x-coordinates of body stations. |
| R | = | Array of body radii at body stations. |
| FD | = | Array of body surface first derivatives at body stations. |
| THETAB | = | Array of $\theta$-angles of body meridian lines. |
| CPBB | = | Array of pressure coefficients. |
| NFT<br>NLT | = | Index codes. |
| NTHETA | = | Number of body meridian lines. |
| XP<br>ZP | = | x and y coordinates of point about which pitching moment coefficient is to be computed. |
| C | = | Coefficient (axial force, normal force or pitching moment). |
| IRROR | = | Error return |
| | = | 0 — success |
| | ≠ | 0 — error occurred in numerical integration of subroutine SIMUN3 |

SUBPROGRAMS CALLED: SIMUN3

COS (Built-in-function)

ERROR RETURNS: Three types of errors are indicated. If fewer than three points are input to subroutine SIMUN3, IRROR = 1. If subroutine SCAMP4, which is called by subroutine SIMUN3, detects an error, IRROR = 2. If subroutine SIMUN3 detects an error in calculating the integral of the function defined by the input points, IRROR = 3. Otherwise, if the call is successful, IRROR = 0.

RESTRICTIONS: $3 \leq (NLT-NFT + 1) \leq 55$

STORAGE: $491_{10} = 753_8$

SUBJECT: FORTRAN IV Subroutine COMCU

PURPOSE: To fit a composite cubic through n points $(x_i, y_i)$ i.e., a separate cubic between each pair of adjacent points, such that the n - 1 cubics are so determined that each matches its neighbors in function value and in the first two derivatives.

METHOD: Rather than solve simultaneously for the 4(n - 1) cubic coefficients, the approach here is to solve simultaneously for the slopes of the composite cubic at the given n points. Thus a linear system of order n, rather than 4n - 4, is involved. It can be shown that a necessary and sufficient condition for continuity of the second derivative is that:

$$(x_{i+1} - x_i)y'_{i-1} + 2(x_{i+1} - x_{i-1})y'_i + (x_i - x_{i-1})y'_{i+1} =$$

$$\frac{3}{(x_i - x_{i-1})(x_{i+1} - x_i)}\left[(x_i - x_{i-1})^2 (y_{i+1} - y_i) + (x_{i+1} - x_i)^2 (y_i - y_{i-1})\right]$$

for i = 2, 3, ..., n - 1.

This yields n - 2 equations in the n unknowns, $y'_i$, i = 1, 2, ..., n. For the 1st and $n^{th}$ equations of the linear system, the boundary conditions on $y'_1$ and $y'_n$ are used. This has been generalized to permit any combination of a given y' or y'' at the end points, e.g., $y'_1$ and $y''_n$ can be given as the boundary conditions. The second derivative of a cubic through two points can be expressed as a function of the first derivatives and of the given point coordinates as follows:

$$\frac{x_2 - x_1}{2} y''_1 = 3\frac{y_2 - y_1}{x_2 - x_1} - 2y'_1 - y'_2$$

and

$$\frac{x_n - x_{n-1}}{2} y''_n = -3\frac{y_n - y_{n-1}}{x_n - x_{n-1}} + y'_{n-1} + 2y'_n$$

Whether the boundary conditions involve first or second derivatives (or both), and no matter what the spacing of the $x_i$ (so long as the $x_i$ form a strictly monotone sequence), the coefficient matrix of the linear system is

tridiagonal (all elements are zero except on the principal diagonal, the first subdiagonal, and the first superdiagonal). When n is large, a considerable time saving and an enormous storage saving can result if the special structure of this matrix is taken advantage of. Hence this subroutine stores the matrix elements in 4n locations (as opposed to $n^2$ ) and then solves the system.

The actual coefficients of the n - 1 cubics of the composite cubic are not found by COMCU. Since on any subinterval $x_i$ , $x_{i+1}$ a cubic is uniquely determined by the known two points and two slopes, the calling program can find the four coefficients of each cubic independently and may often need to do so for only one of the n - 1 cubics. In any case, the subroutine CUBIC2 specifically finds a cubic, given two points and the slope at each point.

USAGE:

CALL COMCU (DA, DB, S, X, Y, L, M, N, NDA, NDB)

DIMENSION S(400), X(400), Y(400)

| | | |
|---|---|---|
| Input: | X | = Array of x-abscissae of input points. |
| | Y | = Array of y-ordinates of input points. |
| | N | = Number of input points. |
| | NDA | = Order (1 or 2) of derivative at X(1). |
| | NDB | = Order (1 or 2) of derivative at X(N). |
| | DA | = Value of derivative at X(1). |
| | DB | = Value of derivative at X(N). |
| | L | = Code. |
| | | = 1, if single precision is to be used. |
| | | = 2, if double precision is to be used. |
| Output: | S | = Array of first derivatives. |
| | M | = Error return |
| | | = O — success |
| | | ≠ O — error detected |

| | |
|---|---|
| **SUBPROGRAMS CALLED:** | None |
| **ERROR RETURNS:** | If overflow occurred, M = 1. Otherwise, M = 0. |
| **RESTRICTIONS:** | The x-abscissae must form a strictly monotone sequence. $N \leq 400$ |
| **STORAGE:** | $1496_{10} = 2730_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine COMP |
| PURPOSE: | To compute the velocity components at an array of points. |
| METHOD: | An array of points is input, and calls to velocity computation routine are made. |
| USAGE: | DIMENSION GP(3, IC), VLS (4, IC) |

CALL COMP (GP, VLS, IC)

| | | | |
|---|---|---|---|
| Input: | IC | = | Number of points in the GP array. |
| | GP(1, I) | = | x-coordinate of $I^{th}$ point. |
| | GP(2, I) | = | y-coordinate of $I^{th}$ point. |
| | GP(3, I) | = | z-coordinate of $I^{th}$ point. |
| Output: | VLS(1, I) | = | u-velocity component at $I^{th}$ point. |
| | VLS(2, I) | = | v-velocity component at $I^{th}$ point. |
| | VLS(3, I) | = | w-velocity component at $I^{th}$ point. |
| | VLS(4, I) | = | $c_p$-velocity component at $I^{th}$ point. |

| | |
|---|---|
| SUBROUTINE CALLED: | LACKEY |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $89_{10} = 131_8$ |

**SUBJECT:** FORTRAN IV Subroutine CP

**PURPOSE:** To compute the pressures coefficients by either the linear or nonlinear equation, given the velocity components in the x , y , and z directions at each of the panels.

**METHOD:** For the linear pressure coefficient,

$$C_p = -2u$$

and for the nonlinear pressure coefficient,

$$C_p = -2u + \beta^2 u^2 - v^2 - w^2$$

where

$u$ = velocity component in the x-direction.

$v$ = velocity component in the y-direction.

$w$ = velocity component in the z-direction.

$\beta = M^2 - 1$

USAGE: CALL CP (K, NP, XMACH, CPCALC, U, V, W, CPP)

DIMENSION U(NP), V(NP), W(NP), CPP(NP)

Input: K = Code

= 0, for wing panels.

= 1, for body panels.

NP = Number of panels.

XMACH = Mach number.

CPCALC = Calculation option.

= 0., calculation of pressure coefficients to use linear equation.

= 1., calculation of pressure coefficients to use nonlinear equation.

| | | | |
|---|---|---|---|
| | U | = | Array of velocity components in x-direction. |
| | V | = | Array of velocity components in y-direction. |
| | W | = | Array of velocity components in z-direction. |
| Output: | CPP | = | Array of pressure coefficients, |

SUBPROGRAMS CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $156_{10} = 234_8$

SUBJECT: FORTRAN IV Subroutine CRNRB

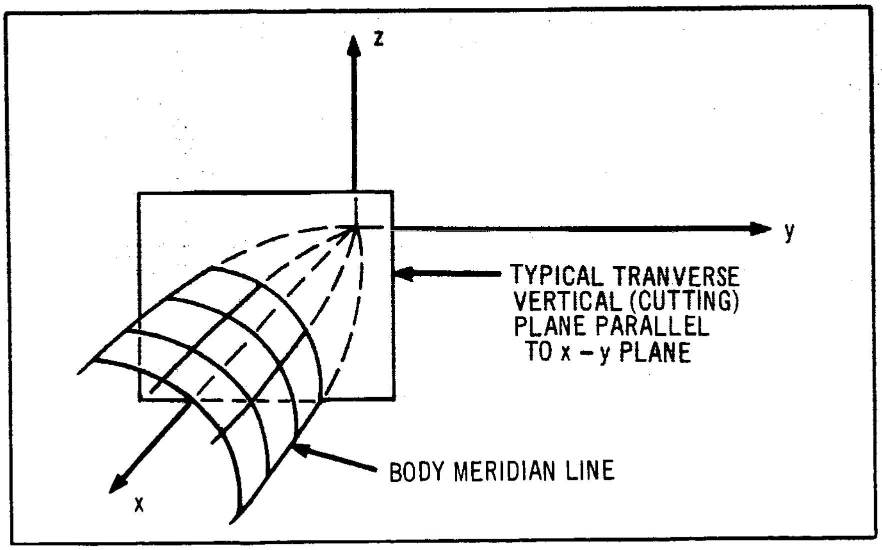
PURPOSE: To calculate body panel (and secondary panel) corner point coordinates.

METHOD: In general, a body panel is defined by four corner points. A program restriction requires either that the panel have streamwise inboard and outboard edges, or that it be subdivided so that its secondary panels have streamwise edges. To satisfy this condition, additional points are located on the panel leading and/or trailing edge.

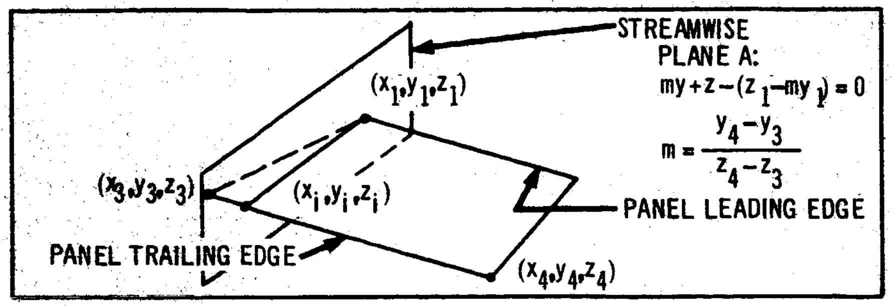
The program calculates the four panel corner points by intersecting with the body meridian lines a series of parallel transverse vertical (cutting) planes. The equations for these planes have the general form,

$$x - c_i = 0,$$

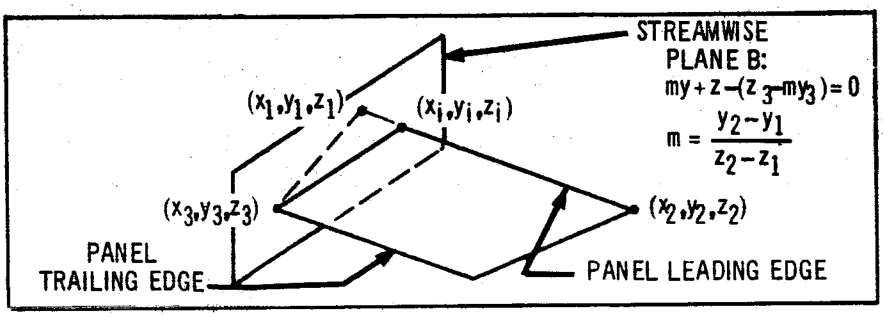
for which $c_i$ is the $i^{th}$ intercept as given on input cards. This is shown in the following sketch:




The program calculates the additional panel points by constructing planes normal to the plane of the panel and through a panel corner point, and intersecting these planes with the panel leading and trailing edges. At most, two such constructions are necessary for each of the two required points. To obtain the first point, the program constructs a streamwise plane (A) normal to the panel trailing edge and through the inboard leading edge corner point and attempts to intersect this plane with the panel trailing edge:

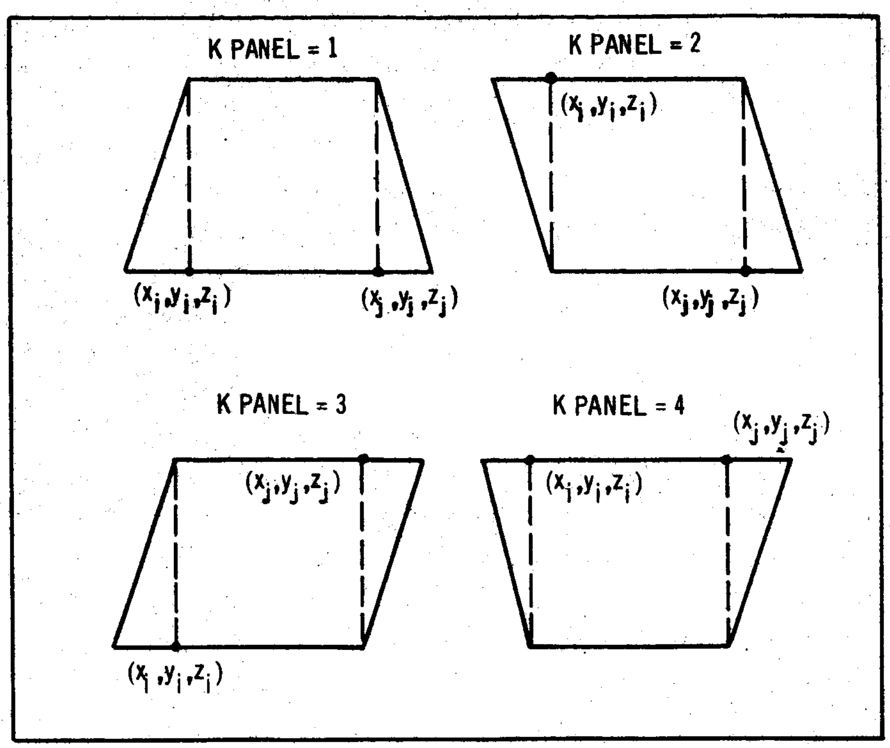



If an intersection with the trailing edge is obtained, this point is defined to be the first of the two required points. Should no intersection be calculated, the program then constructs a streamwise plane normal to the panel leading edge and through the inboard trailing edge corner point and attempts to intersect this plane with the panel leading edge:



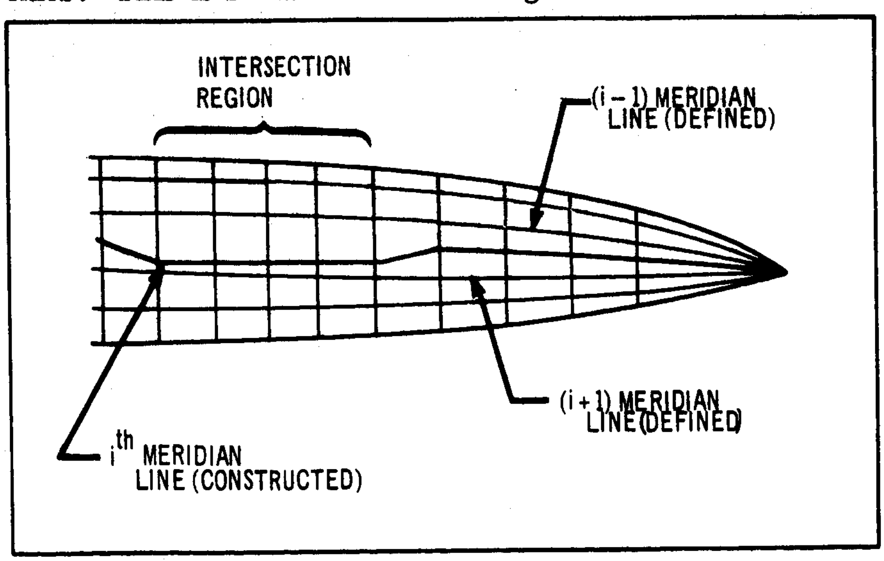

If the previous construction has failed, then this construction yields an intersection, and the point is defined to be the first of the two required points. The calculations used to obtain the second of the two points are identical to those given above except that the planes that are constructed pass through outboard rather than inboard corner points.

As both, either, or neither of these two points can occur on the panel leading (or trailing) edge, four panel configurations are possible, and a code is assigned for each panel. These codes are the following:




These codes are used in later subprograms.

In certain cases the body-wing intersection line does not coincide with a body meridian line, and an additional meridian line is constructed. In the intersection region proper, points on this additional line are defined to be the intersection points; elsewhere, fore and aft of the intersection region, points are obtained by averaging the corresponding points on the two adjacent body meridian lines. This is shown in the following sketch:



USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NWRITE

COMMON /COM1/ (See subroutine PANEL)

Input: KOPTB
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

Output: KSTART
KEND

COMMON /COM2/ (See subroutine BODY)

Input: NPLNB
NPLNW
JLEAD
JTRAIL
NPTS
X
Y
Z
XCEPT
XCEPTW
YCEPTW
ZCEPTW

Output: IMID
KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT

CALL CRNRB

| | |
|---|---|
| SUBPROGRAM CALLED: | POLXN |
| ERROR RETURNS: | Error messages indicate if panel or secondary panel corner points not correctly calculated. |
| STORAGE: | $1282_{10} = 2402_8$ |

SUBJECT: FORTRAN IV Subroutine CRNRW

PURPOSE: To calculate wing panel (and secondary panel) corner point coordinates.

METHOD: In general, a wing panel is defined by four corner points. To satisfy a program restriction that either a panel have streamwise inboard and outboard edges or that it be subdivided so that its secondary panels have streamwise edges, an additional point is determined in the following two cases:

1) For panels on the inboard column if the body and wing intersect in a nonstreamwise line;

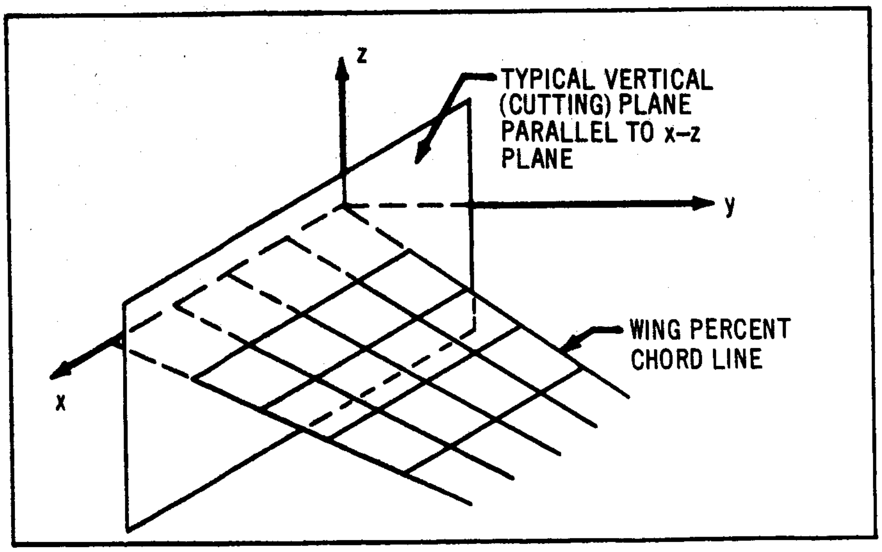
2) For panels on the outboard column if the outboard wing edge is a nonstreamwise edge.

All other panels have streamwise edges and are not subpaneled.

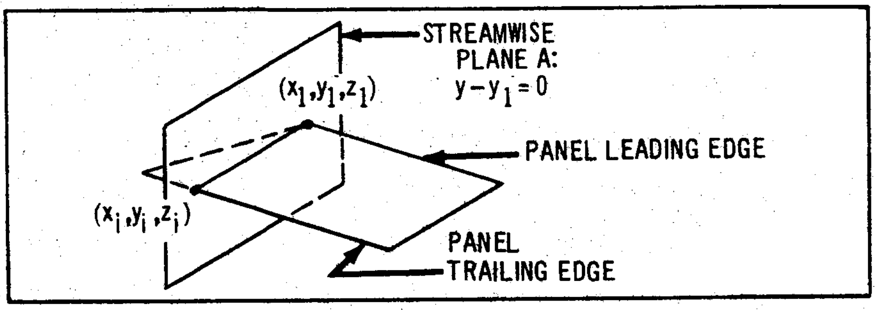
The program calculates the four panel corner points by intersecting with the wing percent chord lines a series of vertical (cutting) planes. The equations for these planes have the general form,

$$y - c_i = 0,$$

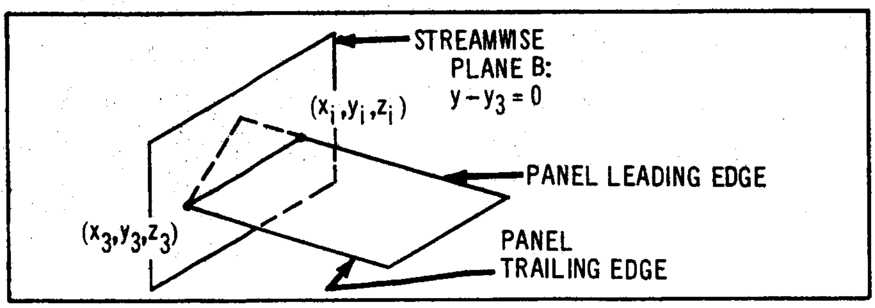
for which $c_i$ is the $i^{th}$ intercept as given on input cards. This is shown in the following sketch.



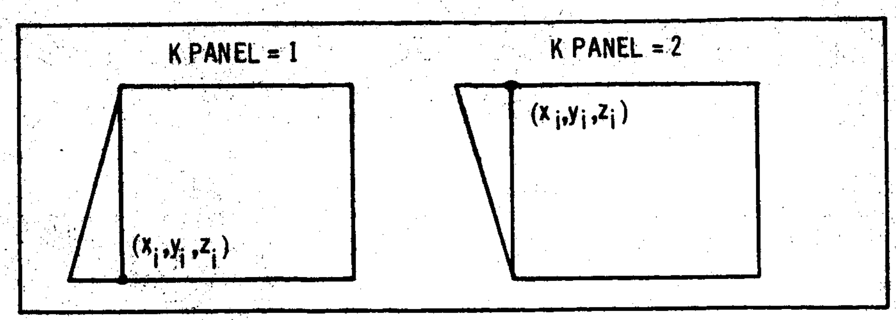
The program calculates the additional panel point, in those cases indicated above, by constructing planes normal to the plane of the panel and through a panel corner point, and by intersecting these planes with the panel leading or trailing edges. Two such constructions, at most, are required to obtain this point. The program first constructs a plane (A) normal to the panel and through a leading edge corner point and attempts to intersect this plane with the panel trailing edge.




If an intersection with the trailing edge is found, it is then defined to be that additional panel point. If no intersection is calculated, the program constructs a plane (B) normal to the panel and through a trailing edge corner point and attempts to intersect this plane with the leading edge.




As this additional point can be on either the leading or trailing edge a code is determined and assigned for any such panel.




USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NWRITE

COMMON /COM1 / (See subroutine PANEL)

Input: KOPTW
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

Output: KSTART
KEND

COMMON /COM2 / (See subroutine WING)

Input: IJ
NPTS
X
Y
Z
KPNT
VALUE
YCEPT
SLOPE

Output: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT

CALL CRNRW

| | |
|---|---|
| **SUBPROGRAM CALLED:** | POLXN |
| **ERROR RETURNS:** | **Error messages indicate if panel (or secondary panel) corner points not calculated.** |
| **STORAGE:** | **$926_{10} = 1636_8$** |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine CUBIC2 |
| PURPOSE: | To fit a cubic to two points, being given the slope at each. |
| METHOD: | The subroutine sets up the system of four simultaneous equations expressing the four given conditions and solves it for the coefficients of the cubic. |
| USAGE: | CALL CUBIC2 (X, Y, D, C, M) |
| | DIMENSION X(2), Y(2), D(2), C(4) |
| | Input: X = Array of x-coordinates. |
| | Y = Array of y-coordinates. |
| | D = Array of first derivatives. |
| | Output: C = Array of cubic coefficients. |
| | M = Error return. |
| | = 1 — success |
| | ≠ 1 — error detected |
| SUBPROGRAMS CALLED: | None |
| ERROR RETURNS: | If M = 2, overflow occurred. If M = 3, X(1) = X(2). Otherwise call is successful and M = 1. |
| RESTRICTIONS: | X(1) ≠ X(2) |
| STORAGE: | $146_{10} = 222_8$ |

**SUBJECT:** FORTRAN IV Subroutine CVEL

**PURPOSE:** To compute the velocity components due to a given pressure difference, given the velocity components resulting from a unit pressure difference across panels.

**METHOD:** The velocity components resulting from a unit pressure difference across the influencing panels are computed and stored on tape previous to calling of this subroutine. Then the velocity components resulting from a given pressure difference across the influencing panels are found by

$$u_i = \sum_{j=1}^{N} U_{ij} \cdot p_j$$

$$v_i = \sum_{j=1}^{N} V_{ij} \cdot p_j$$

$$w_i = \sum_{j=1}^{N} W_{ij} \cdot p_j$$

where $U_{ij}$, $V_{ij}$, and $W_{ij}$ are the velocity components in the x, y, and z directions respectively at panel i induced by panel j, and $p_j$ is the given pressure differences across the influencing panel.

**USAGE:** CALL CVEL(M, N, NT, A, B, C, CL, U, V, W)

DIMENSION A(M), B(M), C(M), U(N), V(N), W(N)

Input:

- M = Number of influenced panels.
- N = Number of influencing panels.
- NT = Logical tape number on which the velocity components due to unit pressure differences are stored.
- A = Dummy array used by the subroutine.
- B = Dummy array used by the subroutine.
- C = Dummy array used by the subroutine.
- CL = Array of given pressure differences.

**Output:** U = Array of velocity components in the x-direction.

V = Array of velocity components in the y-direction.

W = Array of velocity components in the z-direction.

**SUBPROGRAM CALLED:** None

**ERROR RETURNS:** None

**RESTRICTIONS:** None

**STORAGE:** $129_{10} = 201_8$

**SUBJECT:** **FORTRAN IV Subroutine DATE**

**(See subroutine FDATE)**

SUBJECT: FORTRAN IV Subroutine DCPD

PURPOSE: To compute the pressure difference across the wing panels.

METHOD: The pressure difference across a wing panel i is given by:

$$p_{W_i} = \sum_{j=1}^{NM} A_{R_{ij}}^{-1} \cdot n_{W_i}$$

where $A_R^{-1}$ = The inverse of the reduced aerodynamic influence coefficients matrix.

$n_W$ = The velocity component of the wing, normal to the x-axis.

USAGE: CALL DCPD (NM, NTAPEX, A, ALPHAM, CLM)

DIMENSION A(NM), ALPHAM(NM), CLM(NM)

Input:

NM = Number of wing panels.

NTAPEX = Logical tape number, of which $A_R^{-1}$ is stored.

A = Dummy array used by the subroutine.

ALPHAM = Array velocity components on wing panels normal to the x-axis.

Output:

CLM = Array of the pressure difference across wing panels.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE $90_{10} = 132_8$

SUBJECT: FORTRAN IV Subroutine DCPI

PURPOSE: In the presence of a wing, to compute the pressure difference across a body panel.

METHOD: The pressure difference is given by:

$$p_{B_i} = \sum_{j=1}^{NB} A_{BB_{ij}}^{-1} \cdot n_{B_j} + \sum_{j=1}^{NW} E_{ij} \cdot p_{W_i}$$

where $A_{BB}^{-1}$ = The inverse of a partition of the aerodynamic influence coefficients resulting from the influence of the body on the body.

$n_B$ = The velocity component normal to the x-axis, on the body.

E = The matrix formed by the product,

$$A_{BB}^{-1} \cdot A_{BW}$$

where $A_{BW}$ = The partition of the aerodynamic influence coefficients matrix resulting from the influence of the wing on the body.

$p_W$ = The pressure difference across the wing panels.

USAGE: CALL DCPI (NB, NW, NTAPEX, A, ALPHAB, CLW, CLB)

DIMENSION A(NW), ALPHAB(NB), CLW(NW), CLB(NB)

Input:

NB = Number of body panels.

NW = Number of wing panels.

NTAPEX = Logical tape number on which $A_{BB}^{-1}$ and E are stored.

A = Dummy array used by the subroutine.

ALPHAB = Array of velocity components on the body, normal to the x-axis.

CLW = Array of the pressure difference across wing panels.

Output:

CLB = Array of the pressure difference across the body panels.

| | |
|---|---|
| **SUBPROGRAM CALLED:** | None |
| **ERROR RETURNS:** | None |
| **RESTRICTIONS** | None |
| **STORAGE:** | $143_{10} = 217_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine DEFEN1 |
| PURPOSE: | To write records on the Geometry Definition binary tape. |

METHOD:

The format of the binary tape to be written is shown in Appendix D. Each time that DEFEN1 is called it writes two records. Record A consists of ten fixed-point words. Record B contains either nine words or K - J + 1 words, where K and J are given. A ten-word array, IREC, is given. If IREC(3) $\neq$ 0, then record B consists of nine words, which are zeros, and record A has the following format:

| | |
|---|---|
| word 1 | IREC(1) |
| word 2 | 0 |
| word 3 | IREC(3) |
| words 4-10 | 0 |

However, if IREC(3) = 0, then IREC is written as record A and the specified part of a given array is written as record B.

USAGE:

```
DIMENSION   B(1), IREC(10)
CALL DEFEN1 (B, J, K, LTAPE, IREC)
```

Input:

| | |
|---|---|
| B | = B(J) through B(K) is to be written on LTAPE. |
| J | = Subscript. |
| K | = Subscript. |
| LTAPE | = Tape number. |
| IREC | = Array to be written on LTAPE (see METHOD, above). |

| | |
|---|---|
| SUBPROGRAM CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | LTAPE is assumed to be correctly positioned. |
| STORAGE: | $128_{10} = 200_8$ |

SUBJECT: FORTRAN IV Subroutine DEFEND

PURPOSE: To prepare records for the Geometry Definition tape and to monitor tape writing.

METHOD: This subroutine is used by subroutine GEOMD to transfer data from core to tape. If wing-body intersections are not requested, GEOMD calls DEFEND at the end of the case to write records 1-18 (see Appendix D). But if intersections are requested, GEOMD first calls DEFEND to write records 1-10, then finds intersections, and finally calls DEFEND again to write records 11-18.

USAGE:

```
DIMENSION  B(1), LK(3, 9)
LOGICAL  LGDEF(3, 6), LGTAPE
CALL DEFEND (B, LK, LO, LTAPE, LGDEF, LGTAPE,
INTWB)
```

Input: B = Storage buffer for variable-length arrays.

LK = Array that describes contents of B. Consider LK(i, j). The index j refers to a data array as follows:

| j | Data array |
|---|---|
| 1 | Upper wing percent lines |
| 2 | Lower wing percent lines |
| 3 | Upper wing percent lines (part outside body) |
| 4 | Lower wing percent lines (part outside body) |
| 5 | Body meridian lines |
| 6 | Body stations |
| 7 | Wing planform (leading and trailing edge points) |
| 8 | Upper wing-body intersections |
| 9 | Lower wing-body intersections |

A data array is stored starting in B(LK(1, j)). The number of elements

in the data array is LK(3, j). The number of percent lines (j = 1, 2, 3, 4) or meridian lines (j = 5) is LK(2, j). The number of airfoils is LK(2, 7). LK(2, j) is not used for j = 6, 8, 9.

| | | | |
|---|---|---|---|
| | LO | = | Output tape number. |
| | LTAPE | = | Binary tape number. |
| | LGDEF | = | Logical array of problem status (see subroutine GEOMD). |
| | LGTAPE | = | .TRUE. if records are to be written on LTAPE, .FALSE. if not. |
| | INTWB | = | If INTWB < 0, LTAPE is to be rewound and records 1-10 written. If INTWB > 0, records 11-18 and an EOF are to be written. If INTWB = 0, records 1-18 and an EOF are to be written. |
| Output: | LGDEF | = | If LGDEF(1, i) and LGDEF(2, i) are both true, LGDEF(3, i) is set to .TRUE. to indicate that the corresponding data array has been written on LTAPE. |

SUBPROGRAM CALLED: DEFEN1

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $373_{10} = 565_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Function DERIV1 |
| PURPOSE: | To find the first derivative of the quadratic through three given points at a specified one of these points. This provides a good approximation to the slope of a function at a point, particularly if the other two points used are nearby. |
| METHOD: | The subroutine simply finds the unique polynomial of degree two through the given points then evaluates its first derivative at the specified point. |
| USAGE: | D = DERIV1 (X, Y, N)<br><br>DIMENSION X(3), Y(3)<br><br>Input: X = Array of x-coordinates.<br><br>Y = Array of y-coordinates.<br><br>N = Code.<br><br>= 1, 2 or 3 indicating point at which derivative is desired. |
| SUBPROGRAMS CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | The x-coordinates must be distinct, but need not be in any order of evenly spaced. |
| STORAGE: | $129_{10} = 201_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Function DERIV2 |
| PURPOSE: | To find the second derivative of the cubic through four given points ( $x_i$, $y_i$ ) at an arbitrary point whose x-coordinate is given. |
| METHOD: | The subprogram simply finds the unique polynomial of degree three through the given points, then evaluates its second derivative at the desired $x$ , which need not be one of the four given $x_i$ . |
| USAGE: | D = DERIV2 (X, Y, XX)<br>DIMENSION X(4), Y(4)<br>Input: X = Array of x-coordinates.<br>Y = Array of y-coordinates.<br>XX = x-coordinate of point at which second derivative is desired. |
| SUBPROGRAMS CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | The x-coordinates must be distinct but can be in any order and unevenly spaced. |
| STORAGE: | $167_{10} = 247_8$ |

SUBJECT: FORTRAN IV Subroutine DERIVS

PURPOSE: Given a set of n points ( $x_i$ , $y_i$ ), to obtain the first and second derivatives at each of the points.

METHOD: Subroutine SCAMP4 is used to fit a chain of cubics of the general form:

$$y = C_o + C_1x + C_2x^2 + C_3x^3$$

to the set of points and calculate the first derivative at each point from the equation:

$$y' = C_1 + 2C_2x + 3C_3x^2.$$

The second derivative at each point is obtained from the equation:

$$y'' = 2C_2 + 6C_3x.$$

USAGE: CALL DERIVS (XB, R, NBODYS, FD, SD)

DIMENSION XB(NBODYS), R(NBODYS), FD(NBODYS), SD(NBODYS)

| | | | |
|---|---|---|---|
| Input: | XB | = | Array of x-abscissae. |
| | R | = | Array of y-ordinates. |
| | NBODYS | = | Number of points. |
| Output: | FD | = | Array of first derivatives. |
| | SD | = | Array of second derivatives. |

SUBPROGRAM CALLED: SCAMP4

ERROR RETURNS: None

RESTRICTIONS: The array of x-abscissae must form a strictly monotone sequence.

STORAGE: $301_{10} = 455_8$

| | |
|---|---|
| SUBJECT: | FORTRAN Function DISPTA |
| PURPOSE: | To find the distance between two points in N-space. |
| METHOD: | Standard |
| USAGE: | D = DISPTA (P, Q, N)<br>DIMENSION P(N), Q(N)<br>Input: P = Array of coordinates of one point.<br>Q = Array of coordinates of other point.<br>N = Dimension of space.<br>Output: D = Distance between points P and Q. |
| SUBPROGRAM CALLED: | SQRT (Built-in function) |
| ERROR RETURNS: | None |
| RESTRICTIONS: | $N > 0$ |
| STORAGE: | $50_{10} = 62_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine DMAXL |
| PURPOSE: | Given a set of points in 3-space, to find which point is farthest from the line through the end-points. |
| METHOD: | Let $P_1$ and $P_N$ be the first and last points, and let $\bar{u}$ be a unit vector from $\overline{P}_1$ to $\overline{P}_N$. For each intermediate point $P_i$, find the vector $\bar{v} = \overline{P}_i - \overline{P}_1$. Then the distance from $P_i$ to the line $P_1 P_N$ is the magnitude of the cross-product of $\bar{u}$ and $\bar{v}$ The point $P_j$ farthest from the line is selected. If there is more than one such point, the first is selected. |
| USAGE: | DIMENSION P(3, N), U(3) |

CALL DMAXL (P, N, DIST, J, U, NU)

| | | |
|---|---|---|
| Input: | P | = Array of points. |
| | N | = Number of points in P. |
| Output: | DIST | = Distance from point $P_j$ to the line $P_1 P_N$. |
| | J | = Subscript of point $P_j$ (that is, the desired point is P(i, J), i = 1, 3). |
| | U | = Unit vector directed from $P_1$ to $P_N$. |
| | NU | = Error indicator, which is zero for success. |

| | |
|---|---|
| SUBPROGRAMS CALLED: | UVECN<br>VCROS |
| ERROR RETURNS: | NU = 1 if N = 2,<br>= -1 if N < 2,<br>= -2 if $P_1$ and $P_N$ are coincident. |
| RESTRICTIONS: | None |
| STORAGE: | $139_{10} = 213_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine ELLIPR |
| PURPOSE: | Given the horizontal and vertical semi-axes of a rectangular ellipse and the angle of a radius vector to find the length of the vector and the corresponding point on the ellipse. |
| METHOD: | The sine and cosine of the angle (measured in a clockwise direction from the + y axis) are given. The required quantities are computed in a standard manner. |
| USAGE: | CALL ELLIPR (A, B, SIN, COS, RHO, X, Y, NU) |

Input:
- A = Horizontal semi-axis.
- B = Vertical semi-axis
- SIN = Sine of angle.
- COS = Cosine of angle.

Output:
- RHO = Length of radius vector.
- X = x-coordinate.
- Y = y-coordinate.
- NU = Error code, which is zero if successful.

| | |
|---|---|
| SUBPROGRAM CALLED: | SQRT (Built-in function) |
| ERROR RETURNS: | NU = 1 if either of the semi-axes is zero.<br>NU = 2 if the radius vector has zero length. (This indicates that SIN = COS = 0.) |
| RESTRICTIONS: | None |
| STORAGE: | $83_{10} = 123_8$ |

**SUBJECT:** FORTRAN IV Subroutine ENRYCH

**PURPOSE:** Given a set of points that lie on a smooth curve, to interpolate additional points so spaced that between each pair of points the distance from the chord to the assumed interpolation curve is less than a given tolerance.

**METHOD:** For each set of four consecutive points, subroutine OPTIM3 is called to interpolate additional points. Because OPTIM3 measures error between a chord and arc in a y direction, x and y are reversed for each set of four consecutive points if the range of y is greater than the range of x. Also, signs are reversed if necessary to make the equivalent $x_i$ strictly increasing before calling OPTIM3. The inverse operations are performed after OPTIM3 does the interpolation.

**USAGE:**

DIMENSION A(NI), B(NI), C(NO), NU(3)

CALL ENRYCH (A, B, NI, EPS, C, NO, NU)

Input:

- A = Array of independent variable coordinates.
- B = Array of dependent variable coordinates.
- NI = Number of pairs, $A_i$, $B_i$.
- EPS = Arc-chord tolerance.
- NO = Length of array C.
- NU(2) = Logical output tape number on which to write error comments, if any.
- NU(3) = Error message limiter and counter. If an error occurs for which a message is normally written, NU(3) is first reduced by one. The message is then written only if both NU(2) and NU(3) are greater than zero.

Output:

- C = Array of $A_i$, $B_i$ pairs interspersed with interpolated pairs. If NI < 4 or EPS $\leq$ 0, no interpolation takes place; then C contains $A_i$, $B_i$, NO = 2*NI, and NU = 0.
- NO = Number of elements in the C array.
- NU = 0 — success.
  = 2 — at least one set of four consecutive points ($A_i$, $B_i$) is not strictly monotonic in either x or y.

= 3 — more than 199 interpolated points are required between two consecutive points ($A_i$, $B_i$) in order to satisfy the given value of EPS. This may be due to (1) a very small value of EPS, or (2) a "wild" interpolating cubic between two points (points too "rough" or too far apart in that region).

= 4 — C array is too small.

= 5 — the input value of NO is less than 2*NI.

SUBPROGRAM CALLED: OPTIM3

ERROR RETURNS: If an error is encountered in any interpolating interval (NU = 2 or 3), an error message is written if NU(2) and NU(3) > 0, no points are interpolated in that interval, and the routine continues. The value of NU returned is that of the last error encountered (if any).

RESTRICTIONS: For every set of four consecutive points, either the independent variable coordinates or the dependent variable coordinates must be strictly monotonic. Since OPTIM3 uses cubic or quadratic interpolation, some input arrays may cause rather extreme swings in the output array. This may usually be corrected by supplying more input points in the swing region. In general, the output array of points will lie on a curve that maintains first derivative continuity. However, this may not be true if an x,y reversal (see METHOD, above) is required for some but not all sets of four points; appropriate scaling of one variable before entry (preferably by a power of 2) will ensure that either no set or all sets of four consecutive points will be reversed.

STORAGE: $975_{10} = 1717_8$

SUBJECT: FORTRAN IV Subroutine EVAL

PURPOSE: To compute the matrix of the aerodynamic influence coefficients, the velocity components due to a surface distribution of vorticity, and the velocity components due to a surface distribution of sources in the reference plane of the wing.

METHOD: The equations in this subroutine are given in detail in the Appendix B of Part I.

USAGE: CALL EVAL (NI, XCPT, THKW)

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

COMMON/BLOCK/ALPHAS (210), AREA (210), A (210), ALPHAC (110), ALPHAT (110), CHORD (210), ISYM, NPART (210), NPANEL, NROW (2), THETA (210), TAIL, U (210), V (210), VPM (210), VV (210), VPMM (210), W (210), WPM (210), WW (210), WPMM (210), X (210, 3, 4), XBAR (210), XC (210), Y (210, 3, 4), YBAR (210), YC (210), Z (210, 3, 4), ZBAR (210), ZC (210)

Input:

| | | | |
|---|---|---|---|
| | NI | = | Index that indicates start of influencing panels. |
| | XCPT | = | Array of panel control point coordinates. |
| | THKW | = | 0., indicates the aerodynamic influence coefficients and velocity components due to vortices are to be computed. |
| | | ≠ | 0. indicates the velocity components due to wing sources are to be computed. |
| | XMACH | = | Mach number. |
| | SYM | = | Configuration symmetry condition. |
| | KACE | = | 1, indicates body-alone case. |
| | | = | 2, indicates wing-alone case. |
| | | = | 3, indicates wing-body combination case. |

NBODY = Number of body panels.

NWING = Number of wing panels.

NPANEL = Total number of panels.

NTAPEA = Logical number of tape on which the aerodynamic influence coefficients or normal velocity components on the body due to wing sources are written.

NTAPEB = Logical number of tape on which the velocity components are written.

X = Array of x-coordinates of the panel corner points.

Y = Array of y-coordinates of the panel corner points.

Z = Array of z-coordinates of the panel corner points.

THETA = Array of $\theta$ angles (angle between the plane of the panel and a plane parallel to the x-y plane).

ALPHAS = Array of the panel slopes, $\frac{dz}{dx}$

Output: IF THKW = 0.

A = Matrix of the aerodynamic influence coefficients.

U = Velocity components in the x-direction due to vortex distribution.

V = Velocity components in the y-direction due to vortex distribution.

W = Velocity components in the z-direction due to vortex distribution.

If THKW $\neq$ 0.

A = Normal velocity components on the body due to wing sources.

| | | |
|---|---|---|
| U | = | Velocity components in the x-direction due to wing sources. |
| V | = | Velocity components in the y-direction due to wing sources. |
| W | = | Velocity components in the z-direction due to wing sources. |

**SUBROUTINES CALLED:**

$\left.\begin{array}{l}\text{ACOS}\\ \text{TAN}\\ \text{ALOG}\\ \text{SQRT}\\ \text{COS}\\ \text{SIN}\end{array}\right\}$ (Built-in functions)

**ERROR RETURNS:** None

**RESTRICTIONS:** None

**STORAGE:** $3337_{10} = 6411_8$

SUBJECT: FORTRAN IV Subroutine EVAL1

PURPOSE: To compute velocity components due to a surface distribution of sources in the reference plane of the wing.

METHOD: The equations in this subroutine are given in detail in Appendix B of Part I.

USAGE: COMMON/FLOV1/KACE, NPANEL, NBODY, NWING, NBODYS, NROW (2), XMACH, SYM

COMMON/FLOV2/X(210, 3, 4), Y(210, 3, 4), Z(210, 3, 4), NPART(210), ALPHAS(210), THETA(210), XBB(50), R(50), WT(120), T(50), TC(50), SST(210), CHORD(210)

CALL EVAL1 (NI, THKW, XB, YB, ZB, U, V, W, NSTOP)

DIMENSION C(4), Q(4), CPM(4), RPM(4), RTMM(4), CPMM(4), BBETAM(4)

Input:

NI = Index to indicate starting point of influencing panels.

NSTOP = Index to indicate end point of influencing panels.

THKW = 0. velocity components due to vorticies to be calculated.

= 1. velocity components due to sources to be computed.

XB = x-coordinate of field point.

YB = y-coordinate of field point.

ZB = z-coordinate of field point.

NPANEL = total number of panels.

SYM = 0. nonsymmetric configuration

= 1. symmetric configuration

XMACH = Mach number.

X = array of x-coordinates of panel corner points.

Y = array of y-coordinates of panel corner points.

| | | | |
|---|---|---|---|
| | Z | = | array of z-coordinates of panel corner points. |
| | NPART | = | array containing the number of parts on each panel. |
| | THETA | = | array of $\theta$-angles (angle between the plane of the panel and a plane parallel to the x-y plane. |
| | ALPHAS | = | array of panel slopes, $\frac{dz}{dx}$. |
| | SST | = | array of panel vortex strengths. |
| | WT | = | array of panel source strengths. |
| Output: | U | = | u-component of perturbation velocity. |
| | V | = | v-component of perturbation velocity. |
| | W | = | w-component of perturbation velocity. |

SUBPROGRAMS CALLED:

ACOSH
ALOG
SIN
TAN
SQRT

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $2603_{10} = 5053_8$

SUBJECT: FORTRAN IV Subroutine FASNCS

PURPOSE: To compute the arc sine and/or the arc cosine of a normalized floating point argument.

METHOD: The arc sine and arc cosine are calculated as follows:

1) for $0 \leq x \leq 1$, arc cos $x = (1 - x)^{1/2}\, a(x)$

2) for $-1 \leq x < 0$, arc cos $x = \pi - (1 - |x|)^{1/2}\, a(|x|)$

3) for $0 \leq x \leq 1$, arc sin $x = \pi/2 - (1 - x)^{1/2}\, a(x)$

4) for $-1 \leq x < 0$, arc sin $x = -\pi/2 + (1 - |x|)^{1/2}\, a(|x|)$

where a(x) is approximated by an eighth degree polynomial. For x from 1 to -1, the subroutine gives the arc sine in radians from $\pi/2$ to $-\pi/2$, and the arc cosine in radians from 0 to $\pi$.

USAGE: The subroutine FASNCS has two entry points, ASIN and ACOS; the function names ASIN (X) and ACOS (X) are used in FORTRAN arithmetic expressions.

SUBPROGRAM CALLED: SQRT (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: $|x| \leq 1$. If $|x| > 1$, then an error message is printed. The subroutine is accurate to 8 decimal places, with the following exceptions in the arc sine:

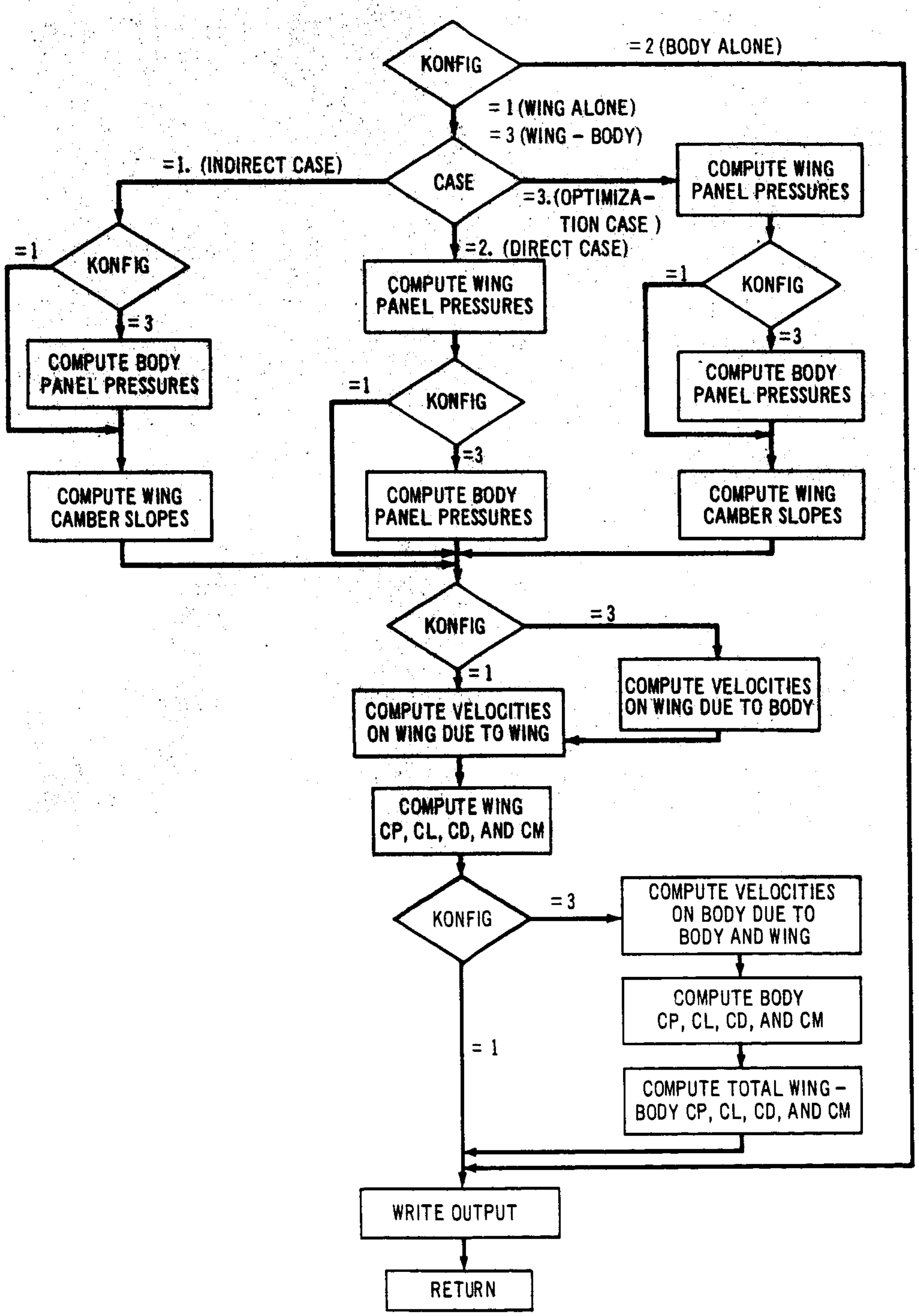
| | |
|---|---|
| $\lvert x \rvert < 0.0015$ | 6 decimal figures |
| $0.0015 \leq \lvert x \rvert < 0.002$ | 7 decimal figures |

STORAGE: $109_{10} = 155_8$

SUBJECT: FORTRAN IV Subroutine FCALC

PURPOSE: To compute pressures, forces, and moments.

METHOD: This subroutine is called by the subroutine FORCES to compute the pressures, forces and moments for the different cases of wing alone, body alone, and wing-body configurations.



USAGE: CALL FCALC

COMMON DATE (2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

COMMON/AVAR/A (210), ACB (21), ABX (100), AWX (110), AREA (210), ARN (2), ARWT (20), ALPHA (210), ALPHAB (210), ALPHAS (210), ALPHAT (110), ALPHAW (110), ALPHAX (110), AWX (110), ALPHAA, ALPHAD, ARA, ARADEG, ARB, ARW, ARAS, AT, AAA

COMMON/BVAR/B (210), BBCL, BBCD, BBCM

COMMON/CVAR/C (210), CHORD (210), CL (210), CPBB (55, 10), CPNN (55, 10, 2), CLS (210), CDR, CASE, CPCALC, CBAR, CONSNT, CLBAR, CLX, CLM, CDM

COMMON/DVAR/D (210), DZDXB (55), DADEG, DARAD

COMMON/IVAR/IPOLAR

COMMON/KVAR/KASE, KONFIG, KPOLAR

COMMON/NVAR/NFMT (9), NROW (2), NS, NPANEL, NACEL, NROWB, NROWW, NCOLW, NBODYS, NTHETA, NTHETS, NXLE, NRG, NPOLAR, NCLX

COMMON/PVAR/POLAR

COMMON/RVAR/R (55), RN (55, 2), RATIOX, RFAREA

COMMON/SVAR/SEMIS, SLC

COMMON/TVAR/TITLE (12), THETA (210), THETAB (10), THETAS (9), THETAN (10, 2), TCL (10), TCD (10), TCM (10), THICK, TWIST

COMMON/UVAR/UBWT (100), UWBT (110), UWWT (110), UNCL (210, 2)

COMMON/VVAR/VBWT (100), VWBT (110), VWWT (110), VNCL (210, 2), VOUT

COMMON/WVAR/WBWT (100), WWBT (110), WWWT (110), WNCL (210, 2)

COMMON/XVAR/XBAR (210), XC (210), XNI (2), XNXN (2), XNTN (2), XB (55), XN (55, 2), XNNCD (2), XNNCL (2), XNNCM (2), XYZ (3), XCL (10), XCD (10), XNACEL, XP, XCPBAR

COMMON/YVAR/YBAR (210), YC (210), YNI (2)

COMMON/ZVAR/ZBAR (210), ZC (210), ZNI (2), ZDELTA (55), ZDN (55, 2), ZP, ZA

Output: Body-alone case

BBCD = Body coefficient of drag.

BBCL = Body coefficient of lift.

BBCM = Body coefficient of pitching moment.

CPBB = Body pressure coefficients.

Wing-alone case

WCD = Wing coefficient of drag.

WCL = Wing coefficient of lift.

WCM = Wing coefficient of pitching moment.

WSCD = Spanwise distribution of wing drag.

WSCL = Spanwise distribution of wing lift.

CPU = Upper surface wing pressure coefficient.

CPL = Lower surface wing pressure coefficient.

CP = Wing pressure difference.

= CPL -CPU

ALPHAU = Upper surface wing panel slopes.

ALPHAL = Lower surface wing panel slopes.

CHORD = Wing panel chord lengths.

Wing-body combination case

Note: In addition to the output given above for the body-alone and wing-alone case, the following output of incremental pressures, forces, and moments on the body due to the wing are given for this case:

| | | |
|---|---|---|
| BWCD | = | Incremental coefficient of drag. |
| BWCL | = | Incremental coefficient of lift. |
| BWCM | = | Incremental coefficient of pitching moment. |
| CPB | = | Increment pressure coefficients. |

and also for the total wing-body configuration:

| | | |
|---|---|---|
| WBCD | = | Total coefficient of drag. |
| WBCL | = | Total coefficient of lift. |
| WBCM | = | Total coefficient of pitching moment. |

SUBROUTINES CALLED:

CAMBW
DCPI
CAMBWB
DCPD
OPTMW
OPTMWB
CVEL
RITE
CP
INTPOL
CAMBER
WLDM
BLDM
OUTB
OUTW
FSF
SQRT (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: NBODY $\leq$ 100
NWING $\leq$ 110

STORAGE: $5440_{10} = 12500_8$

SUBJECT: FORTRAN IV Subroutine FDATE

PURPOSE: To make the date of the run available for printout.

METHOD: The FORTRAN monitor places the current date in core as a BCD word. The first two characters are the month, the second two characters are the day, and the last two characters are the year. This date is printed at the top of each page of a listing.

USAGE: CALL DATE(A, B)

The date, in BCD, is stored and arranged as follows:

Location A = XXXbYY

Location B = ,b19ZZ

where XXX represents a three-character abbreviation for the month, b represents a blank, and YY and ZZ represent the day and year, respectively.

DATE may be used as a function. If so used, the BCD word XXYYZZ is left in the accumulator upon exit from the subroutine,

C = DATE(A, B).

SUBPROGRAM CALLED: None

ERROR RETURNS: None

STORAGE: $41_{10} = 51_8$

SUBJECT: FORTRAN IV Subroutine FFSF

PURPOSE: To enable the FORTRAN program to move past end-of-file marks on FORTRAN written binary tapes, and to skip entire files.

METHOD: The first two words of each record on the designated logical tape unit are read using the IOCS read routine. If the second word is the Hollerith literal "TAPEND," the error code is set equal to the number of files remaining in the file count. Whenever end-of-file is encountered by the IOCS read subroutine, the file count is reduced by one. When the file count reaches zero, the error code is set to zero, and the subroutine returns to the main program. The tape is then positioned just beyond the end-of-file mark, ready to read the first record in the next file. A file count of zero or one will move past one end-of-file mark to the beginning of the next file on the tape. A file count of two will pass two end-of-file marks, thereby skipping the remainder (if any) of the current file and the next file in its entirety.

USAGE: CALL FSF (NFILE, LTAPE, LERROR)

Input: NFILE = The file count described above; the number of files to space forward.

LTAPE = The logical tape unit to be used.

LERROR = Error code. If LERROR = 0, subroutine successful. If LERROR = n, $n > 0$, n indicates the number of unspaced files.

SUBPROGRAMS CALLED: IOCS
FVIO

ERROR RETURNS: None

RESTRICTIONS: If TAPEND indicator was written when the tape was written, FFSF will not read past the TAPEND signal. If not, it is possible to run off the end of the tape.

STORAGE: $52_{10} = 64_8$

**SUBJECT:** FORTRAN IV Subroutine FFSR

**PURPOSE:** To move a designated FORTRAN binary tape forward a specified number of logical records while indicating whether end-of-file was reached.

**METHOD:**

Logical Records (FSR)

The first word in each physical record on the designated tape unit is examined using the IOCS read routine, and if the address is nonzero, the specified record count is reduced by one. If an end-of-file is encountered, the tape is back-spaced so that the end-of-file mark is, in effect, <u>not</u> passed, and the error code is set equal to the number of records remaining to be read. If no end-of-file is encountered, reading continues until the record count reaches zero; zero is stored in the error code and the subroutine returns to the calling program. No redundancy checking is done.

Physical Records (FSPR)

The method is the same as for a logical record except that the record count is reduced by each physical record spaced forward and the first word of each record is not examined.

**USAGE:**

CALL FSR (N, LTAPE, LERROR)

| | | | |
|---|---|---|---|
| Input: | N | = | Number of FORTRAN logical records to forward space. Zero means no spacing. |
| | LTAPE | = | Logical tape unit. |
| | LERROR | = | Error code. |

CALL FSPR (NP, LTAPE, LERROR)

| | | | |
|---|---|---|---|
| Input: | NP | = | Number of physical records to forward space. Zero means no spacing. |
| | LTAPE | = | Logical tape unit. |
| | LERROR | = | Error code. |

**SUBPROGRAMS CALLED:** IOCS
FVIO

RESTRICTIONS: This routine will not proceed beyond an end-of-file mark on tape.

STORAGE: $53_{10} = 65_8$

SUBJECT: FORTRAN IV Subroutine FLOOUT

PURPOSE: To provide optional output of perturbation velocity components and pressure coefficients due to the various singularities — wing pressure singularities, body pressure singularities, wing source singularities, and body line sources and doublets — at points in the near flow field.

METHOD: The velocity contributions are written on four different scratch tapes; one for each type by subroutine LACKEY. Depending upon the type of case — wing along, wing-body or body alone — the velocity components are read in and the pressure coefficients are calculated according to the CPCALC option (see subroutine LACKEY).

Appropriate titles are printed defining which singularities' contributions are being presented. The x, y, z coordinates of the point in question, followed by the u, v, w, velocity components and the pressure coefficients due to that singularity are written on the output tape.

USAGE:

```
COMMON /FLOV4/ COSARA, SINARA, XMACH2, BT2
COMMON /FLOV5/ N1, N2, N3, N4
COMMON /THICK/ THKW, ARA, CPCALC, CAMN, POP
COMMON /INTGA/ GP(3,500), VLS(4,500)
```

CALL FLOOUT (IC)

| | | | |
|---|---|---|---|
| Input: | IC | = | number of points and velocity components to be output. |
| | COSARA | = | cosine of angle of attack |
| | SINARA | = | sine of angle of attack |
| | XMACH2 | = | (Mach Number)$^2$ |
| | BT2 | = | XMACH2-1 |
| | N1, N2, N3, N4 | = | four logical tape numbers of scratch tapes on which the velocity components have been written. |
| | THKW, ARA, CPCALC, CAMN | = | see subroutine LACKEY for description. |
| | POP | = | 0., special printout not requested. |
| | | $\neq$ | 0., special printout is requested. |

GP = scratch array used for x, y, z coordinates.

VLS = scratch array used for u, v, w, and pressure coefficients.

SUBPROGRAMS CALLED: None

ERROR RETURNS: None

RESTRICTIONS: Due to the limitations on the dimension of the scratch arrays, this routine is restricted to maximum of 500 points in the flow field. If more than 500 points are given, the output is printed for the first 500.

STORAGE: $575_{10} = 1077_8$

SUBJECT: FORTRAN IV Subroutine FLOVIZ

PURPOSE: Control routine of the Flow Visualization section.

METHOD: A control word is read from the input tape to determine the option selected. The control word must be one of GRIDS, STREAMLINES, or POINTS. If the control word is not one of the above, control is returned to the calling routine.

Once a control word is recognized, input of data continues and the appropriate option is selected.

USAGE: CALL FLOVIZ

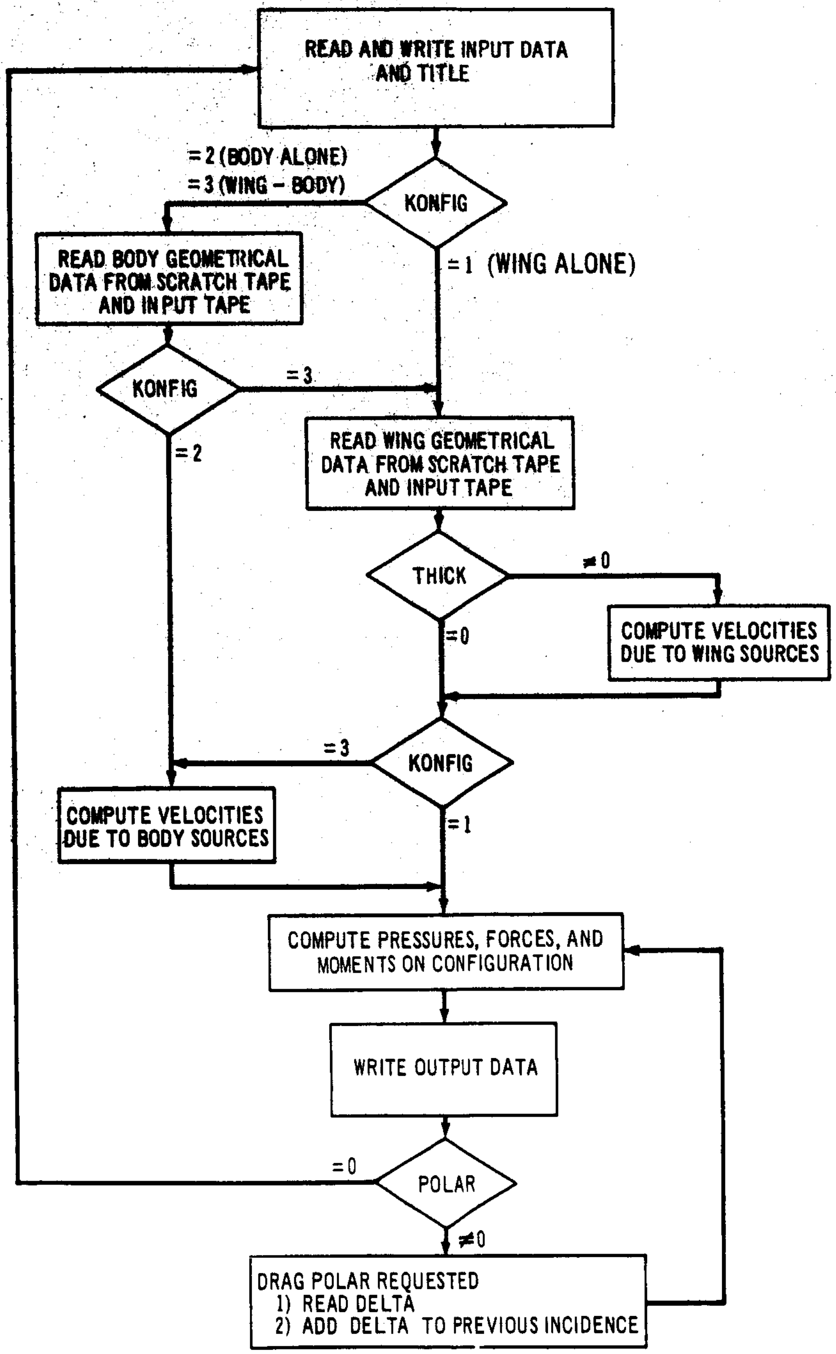
```
DIMENSION X(210, 3, 4), Y (210, 3, 4), Z (210, 3, 4),
NROW (2), NPART (210)

DIMENSION ALPHAS (210), THETA (210)

DIMENSION XBB (50), R (50), WT (120)

DIMENSION T (50), TC (50), SST (210), CHORD (210)

COMMON/FLOV1/KACE, NPANEL, NBODY, NWING,
NBODYS, NWINGS, NROW, XMACH, SYM

COMMON/FLOV2/X, Y, Z, NPART, ALPHAS, THETA,
XBB, R, WT, T, TC, SST, CHORD

COMMON/FLOV4/COSARA, SINARA, XMACH2, BT2

COMMON/FLOV5/N1, N2, N3, N4

COMMON/GRIDA/MODE, DX, DY, DZ, IX, IY, IZ, XO,
YO, ZO, XA, YA, ZA, LIT

COMMON/INTGA/DUM (3500)

COMMON/THICK/THKW, ARA, CPCALC, CAMN, POP

COMMON DAT (4), N8, DUUM (3), N5, N6, DUUUM (5)

DATA GRID/6HGRIDS /, STREAM/6HSTREAM/,
POINT /6HPOINTS/
```

| | |
|---|---|
| SUBPROGRAMS CALLED: | ALLIN<br>GRIDS<br>FLOOUT<br>STRM1<br>LACKEY<br>SQRT |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $908_{10} = 1614$ |

SUBJECT: FORTRAN IV Subroutine FORCES

PURPOSE: Control routine for computing pressures, forces, and moments.

METHOD: The input data cards for a given case are read and other subroutines are called as required to perform the requested calculations.



USAGE: CALL FORCES

COMMON DATE (2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

COMMON/AVAR/A (210), ACB (21), ABX (100), AWX (110), AREA (210), ARN (2), ARWT (20), ALPHA (210), ALPHAB (210), ALPHAS (210), ALPHAT (110), ALPHAW (110), ALPHAX (110), AWX (110), ALPHAA, ALPHAD, ARA, ARADEG, ARB, ARW, ARAS, AT, AAA

COMMON/BVAR/ B (210), BBCL, BBCD, BBCM

COMMON/CVAR/ C (210), CHORD (210), CL (210), CPBB (55, 10), CPNN (55, 10, 2), CLS (210), CDR, CASE, CPCALC, CBAR, CONSNT, CLBAR, CLX, CLM, CDM

COMMON/DVAR/ D (210), DZDXB (55), DADEG, DARAD

COMMON/IVAR/IPOLAR

COMMON/KVAR/KASE, KONFIG, KPOLAR

COMMON/NVAR/NFMT (9), NROW (2), NS, NPANEL, NACEL, NROWB, NROWW, NCOLW, NBODYS, NTHETA, NTHETS, NXLE, NRG, NPOLAR, NCLX

COMMON/PVAR/POLAR

COMMON/RVAR/ R (55), RN (55, 2), RATIOX, RFAREA

COMMON/SVAR/SEMIS, SLC

COMMON/TVAR/TITLE (12), THETA (210), THETAB (10), THETAS (9), THETAN (10, 2), TCL (10), TCD (10), TCM (10), THICK, TWIST

COMMON/UVAR/UBWT (100), UWBT (110), UWWT (110), UNCL (210, 2)

COMMON/VVAR/VBWT (100), VWBT (110), VWWT (110), VNCL (210, 2), VOUT

COMMON/WVAR/WBWT (100), WWBT (110), WWWT (110), WNCL (210, 2)

COMMON/XVAR/XBAR (210), XC (210), XNI (2), XNXN (2), XNTN (2), XB (55), XN (55, 2), XNNCD (2), XNNCL (2), XNNCM (2), XYZ (3), XCL (10), XCD (10), XNACEL, XP, XCPBAR

COMMON/YVAR/YBAR (210), YC (210), YNI (2)

COMMON/ZVAR/ZBAR (210), ZC (210), ZNI (2), ZDELTA (55), ZDN (55, 2), ZP, ZA

Input: See Aerodynamic Card Set of Input Data Format (section 3.3)

SUBPROGRAMS CALLED:

INOUT
READ
BCAM
BTHICK
TVEL
KARMOR
FCALC
FSF
COS (built-in function)

ERROR RETURNS: None

RESTRUCTIONS: NBODY $\leq$ 100
NWING $\leq$ 110

STORAGE: $1020_{10} = 23732_8$

SUBJECT: FORTRAN Subroutine FROOTA

PURPOSE: To find the root of a single-valued function f(x).

METHOD: FROOTA is designed to be called repeatedly until a root is found within a given tolerance. On initial call it is given two points on the function; these two points bracket the root. FROOTA computes two values of x (estimates of the root location) and returns with a code indicating that it wishes to be called again. Calling program finds function values corresponding to the two root estimates and calls FROOTA a second time. FROOTA determines which points now bracket the root, computes new root estimates, and returns. Process continues until FROOTA signals that root has been found.

Root estimates are made as follows: (a) the first root is estimated by assuming the function to be linear between the bracketing points, and (b) the second root is estimated by bisecting the bracketing interval. A root is located when (1) exact root is found, (2) bracketing interval becomes smaller than given tolerance, or (3) root estimate made on the basis of linear interpolation is identical to x-value of bracketing point.

USAGE:

```
   DIMENSION R(10)
   R(9) = 0.
   R(10) = Tolerance
   R(1) = } Initial bracketing values of x
   R(3) = }
   DO α I = 1,50
   R(2) = Function value corresponding to R(1)
   R(4) = Function value corresponding to R(3)
   CALL FROOTA(R, J)
   IF (J) β, γ, α
α  CONTINUE
β  error exit
γ  success exit. Root is in R(1).
```

Input: (First call only. Assume root bracketed by $x_a$, $x_b$ where $x_a > x_b$ or $x_a < x_b$.)

R(1) $= x_a$

R(2) $= f(x_a)$

R(3) $= x_b$

R(4) $= f(x_b)$

R(9) = 0.
R(10) = Tolerance (in x), which may be zero

Input: (2nd and succeeding calls)

R(2) = $f\big(R(1)\big)$

R(4) = $f\big(R(3)\big)$

Output: R(1) = Root estimate if J = 1
= Root if J = 0

R(3) = Root estimate if J = 1

R(5) = $x_c$

R(6) = $f(x_c)$

R(7) = $x_d$

R(8) = $f(x_d)$

where $x_c$ and $x_d$ are values of x which currently bracket the root ($x_c < x_d$).

R(9) = Number of times that FROOTA has been called.

J = 0 The root has been found.
= 1 The root has not yet been located.
= -2 R(9) < 0.
= -3 $f(x_a) * f(x_b) > 0$ on the first call, which means that the root is not bracketed.

SUBPROGRAM CALLED: None

RESTRICTIONS: Function values at limits of initial interval must not have the same sign. Function must be continuous over interval. It may have a number of roots but, of course, FROOTA will only find one. Last six elements of R-array must not be changed between successive calls for a particular case.

STORAGE: $186_{10} = 272_8$

**SUBJECT:** FORTRAN IV Subroutine FSF

(See Subroutine FFSF)

**SUBJECT:** **FORTRAN IV Subroutine FSR**

**(See Subroutine FFSR)**

SUBJECT: FORTRAN IV Subroutine FTAN

PURPOSE: To compute the tangent of a floating point radian argument.

METHOD: The tangent function is computed as follows:

1) if $0 \leq |x| \leq 10^{-4}$ radians, $\tan x = x$

2) if $10^{-4} < |x| \leq \pi/4$ radians,

$$\tan x = \cfrac{x}{1 - \cfrac{x^2}{3 - \cfrac{x^2}{5 - \cfrac{x^2}{7 - \cfrac{x^2}{9 - \cfrac{x^2}{11}}}}}}$$

3) if $\pi/4 < |x| < \pi/2$ radians, and

a) $x > 0$, then $\tan x = \dfrac{1}{\tan(\pi/2 - x)}$

b) $x < 0$, then $\tan x = \dfrac{1}{\tan(\pi/2 - |x|)}$

USAGE: The function name TAN (X) is used in an arithmetic expression where the argument X is a FORTRAN expression.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: X is restricted: $|X| < 2^{26} \cdot \pi$ and not within two bits of an odd multiple of $\pi/2$.

If $|X| > 2^{26} \cdot \pi$ an error message is given.

If the argument is within two bits of an odd multiple of $\pi/2$, an error message is given.

The accuracy is at least 8 significant digits for $0 \leq x \leq \pi$ radians, at least 7 significant digits for $\pi < x \leq 2\pi$ and at least 6 significant digits otherwise.

STORAGE: $124_{10} = 174_8$

SUBJECT: FORTRAN IV Subroutine GEOMD

PURPOSE: To control subprograms that deal with the first phase of wing-body geometry.

METHOD: A buffer is set up (see Appendix A) that is used by all lower-level subroutines for all variable-length storage arrays. Data cards are read by subroutine INTURP until a valid command (in columns 1-6) is found. After any command except DEFEND is executed, data cards are again read until another valid command is encountered; a DEFEND command returns control to the calling program. For a command BODY (or B0DY), subroutine BODY1 is called to read body definition data and compute points on body meridian lines. A command WING (or W1NG) results in a call of subroutine WINGA to read wing definition data and compute points on wing percent lines. A WBX command is used to find the intersections of previously defined wing percent lines with a body surface. The TDUMP command sets a code which will later result in a printout of the contents of the definition tape. A NOTAPE (or N0TAPE) command sets a code to suppress the writing of a definition tape (see Appendix D).

USAGE:

```
DIMENSION  DATE(2)
COMMON /LGEOMD/ LGDEF(3, 6)
LOGICAL  LGDEF
CALL GEOMD (DATE, LI, LO, LER, LTAPE)
```

Input: DATE = Date (alphameric).

LI = Input tape number.

LO = Output tape number.

LER = Output tape number for error comments.

LTAPE = Definition tape number.

Output: LDGEF = Logical array describing which definitions were requested and completed. LGDEF(i, j) is .TRUE. if the $i^{th}$ statement is true for the $j^{th}$ component.

<u>$j^{th}$ component</u>

1. Upper wing surface percent lines.
2. Lower wing surface percent lines.

3. Body surface meridian lines.

4. Wing planform points.

5. Intersections of upper wing percent lines with body surface.

6. Intersections of lower wing percent lines with body surface.

$i^{th}$ statement

1. This component requested.

2. This component successfully computed.

3. This component successfully computed and the results written on LTAPE.

SUBPROGRAMS CALLED:

CLEAR } (See Appendix A)
INIBFR }
TDUMP
DEFEND
BODY1
INTURP
WINGA
WBXUL

ERROR RETURNS: The LGDEF array will reflect any errors detected. Lower-level subroutines write error messages on tape LER.

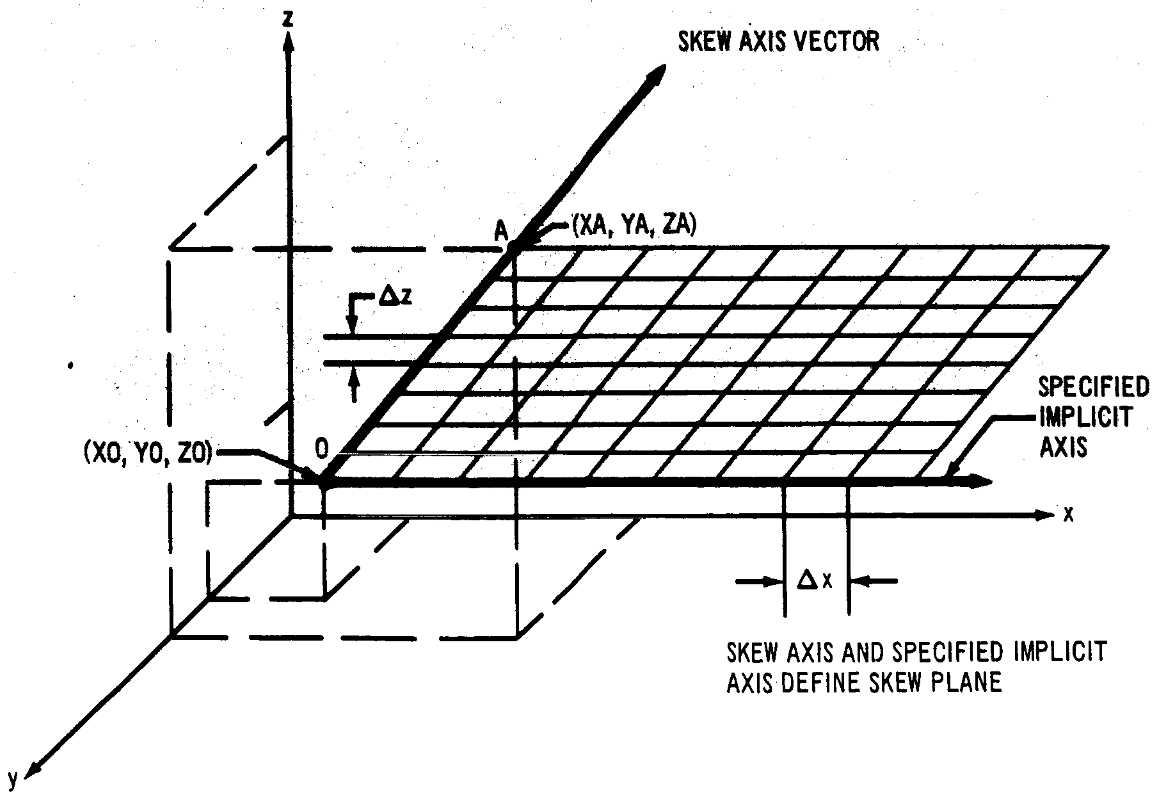
RESTRICTIONS: The suggested order for commands is BODY, WING, WBX, DEFEND. The minimum set of commands is BODY, DEFEND (or WING, DEFEND). The last command must always be DEFEND. NOTAPE must be placed before WBX (if present). BODY, . . . , WING or WING, . . . , BODY must precede WBX. A required set of data cards (see Input Data Format, 3.3) must follow each command except TDUMP, NOTAPE, and DEFEND.

STORAGE: $10289_{10} = 24061_8$

SUBJECT: FORTRAN IV Subroutine GRIDS

PURPOSE: Computes two- and three-dimensional grids, and two-dimensional skew grids. Field velocity components and $C_p$'s are computed at the grid lattice points.

METHOD: The input data described below under USAGE defines the rectangular-grid system. The two-dimensional skew grid is defined by one of the implicit axis (user specified) and by a vector formed by the requested grid origin point, and another input point that may be anywhere in the space of interest. The grid lattice in this case is specified by incremental distances given along an implicit axis and by projections of incremental distances given along one of the implicit axes on the vector-defined axis as shown in the sketch below:



USAGE:

COMMON/GRIDA/MODE, DX, DY, DZ, IX,IY, IZ, XO, YO, ZO, XA, YA, ZA, LIT

COMMON/INTGA/GP (3, NMAX), VLS (4, NMAX)

CALL GRIDS

| | |
|---|---|
| DX | = length of increment in x-direction. |
| DY | = length of increment in y-direction. |
| DZ | = length of increment in z-direction. |
| IX | = number of increments in x-direction. |
| IY | = number of increments in y-direction. |
| IZ | = number of increments in z-direction. |
| XO | = x-coordinate of grid origin. |
| YO | = y-coordinate of grid origin. |
| ZO | = z-coordinate of grid origin. |
| XA | = x-coordinate of skew grid vector point. |
| YA | = y-coordinate of skew grid vector point. |
| ZA | = z-coordinate of skew grid vector point. |
| LIT | = a BCD character x, y, or z, defines the implicit axis to be used in the skew grid computation. |
| MODE | = 1, output as y-z planar cuts. |
| | = 2, output as x-z planar cuts. |
| | = 3, output as x-y planar cuts. |
| | = 4, skew grid. |
| GP (I, J) | = a scratch array used to contain the coordinate of a 2-d planar cut. |
| VLS (I, J) | = a scratch array used to contain the points in the above array. |

Outputs: The coordinates, the actual velocities, a normalized velocity vector, and the $C_p$ at each point of the grid are written on the output tape.

SUBPROGRAM CALLED:

COMP
BLUNDR
FLOOUT
SQRT

ERROR RETURNS: See RESTRICTIONS below.

RESTRICTIONS: The product of the increment parameters for a cut must be less than 500. Grids will not be calculated and an error message is written on the output tape if the above is related.

STORAGE: $1144_{10} = 2214_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine INOUT |
| PURPOSE: | To print out all input aerodynamic data. |
| METHOD: | The input data is written on the output tape for subsequent off-line printing. |
| USAGE: | CALL INOUT (NTAPEO, KASE, CPCALC, POLAR, THICK, VOUT, RFAREA, XP, ZP, XMACH, SYM) |

Input:

| | | |
|---|---|---|
| NTAPEO | = | Logical number of output tape. |
| KASE | = | Code that selects case option. |
| CPCALC | = | Code that selects pressure coefficient calculation option. |
| POLAR | = | Code that selects drag polar option. |
| THICK | = | Code that selects wing thickness option. |
| VOUT | = | Code that selects velocity components printout option. |
| RFAREA | = | Half wing reference area. |
| XP | = | x-coordinate of point about which the pitching moments are to be computed. |
| ZP | = | z-coordinate of point about which the pitching moments are to be computed. |
| XMACH | = | Mach number. |
| SYM | = | Configuration symmetry condition. |

| | |
|---|---|
| SUBPROGRAM CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $472_{10} = 730_8$ |

SUBJECT: FORTRAN IV Subroutine INPUTB

PURPOSE: To read body definition and transformation data from scratch tapes as provided by previous program section and to read body paneling data from input cards.

METHOD: Program uses direct read tape statements to obtain body definition and transformation from scratch tapes (see Appendix D for tape format). Similarly program reads input tape directly for body paneling input card data.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: ND1
ND4
NREAD
NWRITE

COMMON /COM1/ (See subroutine PANEL)

Output: KOPTB
KINT
XI
YI
ZI
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine BODY)

Output: NPLNB
NPLNW
JLEAD
JTRAIL
NPTS
X
Y
Z
XCEPT
XCEPTB
XCEPTW
YCEPTW
ZCEPTW
CODEBW

CALL INPUTB

CALLED: FSR
CHECK

ERROR RETURNS: Error message indicates whether body cutting planes are incorrectly defined on input cards. Error messages also indicate whether size of subscripted array is exceeded by input card data or by data on scratch tapes.

STORAGE: $5972_{10} = 13524_8$

SUBJECT: FORTRAN IV Subroutine INPUTW

PURPOSE: To read wing definition and transformation data from scratch tape as provided by previous program section and to read wing paneling data from input cards. Program reads input cards that define wing cutting planes, and, if an option is used, program reads actual upper and lower airfoil surface coordinates from which wing camber and thickness are calculated.

METHOD: Program uses standard read tape instructions to obtain wing definition from scratch tapes (see Appendix D for tape format). Similarly program reads input tape for wing paneling card data.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: ND4
NREAD
NWRITE

COMMON /COM1/ (See subroutine PANEL for description)

Output: KOPTW
KOPTF
NUMS
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING for description)

Output: NPTS
X
Y
Z
KPNT
VALUE
YCEPT
SLOPE

CALL INPUTW

SUBPROGRAMS CALLED: FSR
CHECK

ERROR RETURNS: Error messages indicate if size of subscripted array is exceeded by data on input cards or on scratch tapes.

STORAGE: $5946_{10} = 13472_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine INTAPE |
| PURPOSE: | To input geometrical data to the aerodynamic links of the program. |
| METHOD: | The geometrical data are read into the aerodynamic links of the program from binary scratch tapes generated in the geometry links. The format of the tape that transfers the panel geometry from the paneling links is described in Appendix D. The geometry of the isolated body is transferred from the transformation links. The format of this tape is given in Appendix D. |
| USAGE: | CALL INTAPE |

COMMON DATE (2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

COMMON/BLOCK/ALPHAS (210), AREA (210), A (210), ALPHAC (110), ALPHAT (110), CHORD (210), ISYM, NPART (210), NPANEL, NROW (2), THETA (210), TAIL, U (210), V (210), VPM (210), VV (210), VPMM (210), W (210), WPM (210), WW (210), WPMM (210)

DIMENSION X (210, 3, 4), XBAR (210), XC (210), Y (210, 3, 4), YBAR (210), YC (210), Z (210, 3, 4), ZBAR (210), ZC (210)

| Input: | | |
|---|---|---|
| | NBODY | = Number of body panels. |
| | NWING | = Number of wing panels. |
| | NTAPEB | = Logical tape number from which isolated body geometry data is read. |
| | NTAPEC | = Logical tape number from which panel geometry data is read and on which panel geometry data is rewritten in a different format for use in the aerodynamics links. |
| | NTAPED | = Logical tape number from which additional isolated body geometry data is read. |

| | |
|---|---|
| SUBPROGRAM: CALLED: | None |
| ERROR RETURNS: | None |

| | |
|---|---|
| **RESTRICTIONS:** | NBODY $\leq$ 100 <br> NWING $\leq$ 110 |
| **STORAGE:** | $1260_{10} = 2354_8$ |

SUBJECT: FORTRAN IV Subroutine INTPOL

PURPOSE: The drag of an individual panel may be obtained by multiplying the panel pressure by the product of the average downwash over the panel and the area. The program calculates panel pressure and downwash $(n_i)$ at the control point. As the control point is normally located at 95 percent of the panel chord, the downwash at this point is often greater than the average downwash of the panel. Thus it is necessary to interpolate for the average downwash of the panel in calculating the drag.

METHOD: For the first panel of any chordwise column, the average downwash is obtained by a linear extrapolation of the downwash at the control points of the first two panels.

$$\bar{n}_1 = n_1 + \frac{(\bar{R} - R)(n_2 - n_1)}{1 + R\left(\frac{C_2}{C_1} - 1\right)}$$

where $R$ = The location of the control point as a fraction of the local panel chord.

$\bar{R}$ = The location of the point as a fraction of the local panel chord at which the downwash used in drag calculations is to be found.

$C$ = The local chord length of the panels.

For the remainder of the panels in any chordwise column of panels, the downwash is obtained by a linear interpolation of the downwash of the adjacent panels.

$$\bar{n}_i = n_i + \frac{C_i}{C_{i-1}} \frac{(\bar{R} - R)(n_i - n_{i-1})}{1 + R\left(\frac{C_i}{C_{i-1}} - 1\right)}$$

USAGE: CALL INTPOL (NM, NROW, RATIOX, RATIOD, CHORD, ALPHAM, ALPHAD)

DIMENSION CHORD(NM), ALPHAM(NM), ALPHAD(NM)

Input: NM = Total number of panels on wing or body.

NROW = Number of panels in a chordwise column.

RATIOX = Location of the control point as a fraction of the local panel chord.

RATIOD = Location of the point as a fraction of the local panel chord at which the downwash is to be found.

CHORD = Array of the local chord lengths of the panels.

ALPHAM = Array of panel downwash at the control point locations.

Output: ALPHAD = Array of interpolated panel downwash values.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $177_{10} = 261_8$

SUBJECT: FORTRAN IV Function INTURP

PURPOSE: To read and write a data card, then find which word in a given table is the same as the first word in the data card.

METHOD: Let LI be the input tape and LO the output tape. INTURP reads the next data card on LI with FORMAT (13A6, A2). If LO > 0, the contents of the card are written on LO with FORMAT (1H0,14A6). A fixed-point comparison is made between the first word on the card and each word in the table until a match is found or the end of the table is reached. The function value returned is the position in the table that matched the first word in the data card, or else zero if no match was found.

USAGE:

```
DIMENSION    T (N)
K   = INTURP (T, N, LI, LO)
```

T = Table.
N = Number of words in T.
LI = Input tape.
LO = Output tape.
K = Function value.

EXAMPLE: A data deck might have three types of data cards, each group with a header card having a code word in the first six columns, and perhaps comments in the remaining columns. Assume the code words are BODY, WING, and DEFINE. A portion of the program could be written as follows:

```
DIMENSION    T(3)
DATA T/4HBODY, 4HWING, 6HDEFINE/
10    K = INTURP (T, 3, 5, 6)
      IF (K) 10, 10, 20
20    GO TO (100, 200, 300), K
```

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $98_{10} = 142_8$

SUBJECT: FORTRAN IV Subroutine INVBB

PURPOSE: For the wing-body case, to preprocess the aerodynamic influence coefficients matrix.

$$\begin{bmatrix} A_{BB} & A_{BW} \\ A_{WB} & A_{WW} \end{bmatrix}$$

where

$A_{BB}$ = The influence on the body due to the body

$A_{WB}$ = The influence on the wing due to the body

$A_{BW}$ = The influence on the body due to the wing

$A_{WW}$ = The influence on the wing due to the wing

by inverting $A_{BB}$ and partitioning $A_{WB}$ in preparation for reduction of the matrix (subroutine REDUCE).

METHOD: The portion of the aerodynamic influence coefficients resulting from the influence of the body is read from logical tape NTAPEA by columns into core. $A_{WB}$ is written immediately on logical tape NTAPED. $A_{BB}$ is inverted in core by the single precision inversion subroutine SINVRT. The matrix inverse is then stored on NTAPED behind $A_{WB}$. Both matrices are written by columns as a logical record. Logical tape NTAPED is rewound and NTAPEA is left at the position of the last read instruction.

USAGE: CALL INVBB

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION ABB(115, 115), AWB(110)

Input:

| | | |
|---|---|---|
| NTAPEA | = | Logical number of scratch tape. |
| NTAPED | = | Logical number of scratch tape. |
| NTAPEO | = | Logical number of output tape. |
| NBODY | = | Number of body panels. |
| NWING | = | Number of wing panels. |
| A | = | Aerodynamic influence coefficients matrix on logical tape NTAPEA. |

Output: AWB = Partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the wing written on logical tape NTAPED.

$ABB^{-1}$ = Inverse of the partition of the aerodynamic influence coefficients matrix resulting from the influence of the body on the body written on logical tape NTAPED.

SUBPROGRAM CALLED: SINVRT (see Appendix B)

ERROR RETURNS: If an error occurs in the inverstion of the matrix, the message "ERROR IN INVERSION OF BODY MATRIX" will be written on the output tape. The values for IRR1, IRR2, and SCALE returned by subroutine SINVRT are also written on the output tape.

RESTRICTIONS: The matrix to be inverted must not exceed the order 110.

STORAGE: $13566_{10} = 32402_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine INVRW |
| PURPOSE: | For the wing-body case, to invert the reduced aerodynamic influence coefficients matrix. |
| METHOD: | The reduced aerodynamic influence coefficients matrix is read from logical tape NTAPEE into core. The matrix is inverted in core by the single-precision inversion subroutine SINVRT. The matrix inverse is then stored on logical tape NTAPEE behind original matrix with an end-of-file separating the two matrices. An end-of-file is also written after the inverted matrix. The tape is then rewound. |
| USAGE: | CALL INVRW |

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION ARWW(115, 115)

| | | |
|---|---|---|
| Input: | NTAPEE | = Logical number of scratch tape. |
| | NWING | = Number of wing panels. |
| | ARWW | = Reduced aerodynamic influence coefficients matrix. |
| Output: | $ARWW^{-1}$ | = Inverse of the reduced aerodynamic influence coefficients matrix. |

| | |
|---|---|
| SUBPROGRAM CALLED: | SINVRT (see Appendix B) |
| ERROR RETURNS: | If an error occurs in the inversion of the matrix, the message "ERROR IN INVERSION OF REDUCED MATRIX" will be written on the output tape. The values of IRR1, IRR2, and SCALE returned by subroutine SINVRT are also written on the output tape. |
| RESTRICTIONS: | The matrix to be inverted must not exceed the order 110. |
| STORAGE: | $13427_{10} = 32163_8$ |

SUBJECT: FORTRAN IV Subroutine INVW

PURPOSE: For the wing-alone case, to invert the aerodynamic influence coefficients matrix and to store both the matrix and its inverse on a scratch tape.

METHOD: The aerodynamic influence coefficients matrix previously computed is read from logical tape NTAPEA into core and simultaneously stored on logical tape NTAPEE, where NTAPEA and NTAPEE are logical tape numbers of scratch tapes stored in COMMON. The matrix in core is inverted by a single precision inversion Subroutine SINVRT. The inverse is stored behind the original matrix on logical tape NTAPEE with an end-of-file mark separating the two matrices. An end-of-file mark is also written after the inverted matrix. Both tapes are then rewound.

USAGE: CALL INVW

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION AWW(115, 115)

Input: AWW = Aerodynamic influence coefficients matrix on logical tape NTAPEA.

Output: AWW = Aerodynamic influence coefficients matrix written on logical tape NTAPEE.

$AWW^{-1}$ = Inverse aerodynamic influence coefficients matrix written on logical tape NTAPEE.

SUBPROGRAM CALLED: SINVRT (See Appendix B)

ERROR RETURNS: If an error occurs in the inversion of the matrix the message ERROR IN INVERSION OF WING ONLY MATRIX will be written on the output tape. The values for IRR1, IRR2, and SCALE as returned by subroutine SINVRT are also written on the output tape.

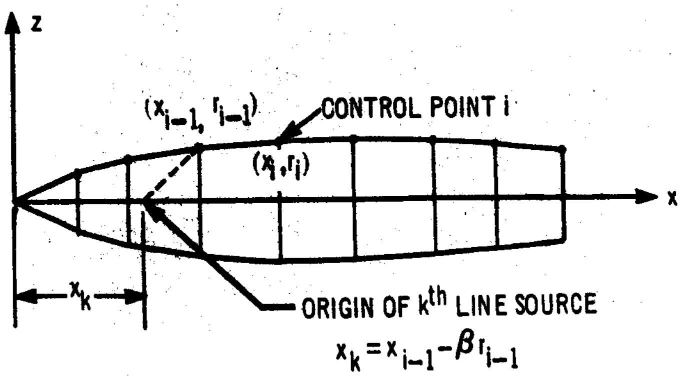
RESTRICTIONS: The matrix must not exceed the order 110.

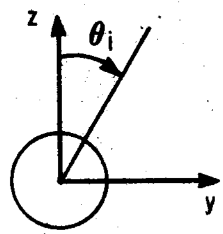
STORAGE: $13467_{10} = 32233_8$

SUBJECT: FORTRAN IV Subroutine KARMOR

PURPOSE: For the isolated body, to compute the velocity components and the resulting pressures, forces, and moments induced by the body sources and doublets.

METHOD: The body, considered to be made up of a series of K parabolic arcs, is defined by a radius r and camber $\Delta z$ at each body station x along the body axis.




A linearly varying source is placed at the nose of the body (i = 1) to give the proper conical tip. The strength of this source is determined by the tangency condition at the nose:

$$T_{S1} = \frac{\tan \delta}{\sqrt{\cot^2 \delta - \beta^2} + \tan \delta \, \cosh^{-1}\left(\frac{\cot}{\beta}\right)}$$

where $\delta$ is the semi-vertex of the nose.

A quadratically varying source, also placed at the nose, satisfies the boundary condition at i = 1:

$$a_{BBS1}\, T_{S1} + a_{BBS2_{11}}\, T_{S2_1} = \left(\frac{dr}{dx}\right)_1 \left[1 + u_{BBS1}\, T_{S1} + u_{BBS2_{11}}\, T_{S2_1}\right]$$

which yields:

$$T_{S2_1} = \frac{\left(\frac{dr}{dx}\right)_1 - \left[a_{BBS1} - \left(\frac{dr}{dx}\right)_1 u_{BBS1}\right] T_{S1}}{a_{BBS2_{11}} - \left(\frac{dr}{dx}\right)_1 u_{BBS2_{11}}}$$

where

$$a_{BBS1_{ik}} = \frac{\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}}{r_i}$$

$$a_{BBS2_{ik}} = \frac{1}{r_i}\left\{(x_i - x_k)\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}\right.$$

$$\left. - \beta^2 r_i^2 \cosh^{-1} \frac{x_i - x_k}{\beta r_i}\right\}$$

$$u_{BBS1_{ik}} = - \cosh^{-1} \frac{x_i - x_k}{\beta r_i}$$

$$u_{BBS2_{ik}} = 2\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}$$

$$- 2 (x_i - x_k) \cosh^{-1} \frac{x_i - x_k}{\beta r_i}$$

$$r_i = \sqrt{y_i^2 + z_i^2}$$

In general, the strength of the quadratically varying source of the $i^{th}$ segment is given by:

$$TS2_i = \frac{\left(\frac{dr}{dx}\right)_i - \left[a_{BBS1_{i1}} - \left(\frac{dr}{dx}\right)_i u_{BBS1_{i1}}\right] T_{S1} - \sum_{k=1}^{i-1}\left[a_{BBS2_{ik}} - \left(\frac{dr}{dx}\right)_i u_{BBS2_{ik}}\right] T_{S2_k}}{a_{BBS2_{ii}} - \left(\frac{dr}{dx}\right)_i u_{BBS2_{ii}}}$$

The end of a closing body presents a special problem since the influence coefficients cannot be evaluated on the axis. The boundary condition on the net source strength at the end of a closing body must be used. The manner in which the strength of the source varies along the axis and the strengths of the previously evaluated sources are known. Therefore, at the end of a closing body, i = NBODYS and $r_i = 0$:

$$(x_{NBODYS} - x_1)\ T_{S1} + \sum_{k=1}^{NBODYS} (x_{NBODYS} - x_k)^2\ T_{S2_k} = 0$$

which yields:

$$T_{S2_{NBODYS}} = \frac{-(x_{NBODYS} - x_1)\ T_{S1} - \sum_{k=1}^{NBODYS-1} (x_{NBODYS} - x_k)^2\ T_{S2_k}}{\left[x_{NBODYS} - (x - \beta r)_{NBODYS-1}\right]^2}$$

It should be noted that the $k^{th}$ line source is located at the distance:

$$x_k = x_{i-1} - \beta r_{i-1}$$

from the nose of the body.

The strengths of the doublets are determined in a similar manner. In terms of the aerodynamic influence coefficients, the boundary condition is written:

$$\left(\frac{dr}{dx}\right)_i \left[u_{BBD1_{i1}}\ T_{D1} + \sum_{k=1}^{K} u_{BBD2_{ik}}\ T_{D2_k}\right] =$$

$$a_{BBD1_{i1}}\ T_{D1} + \sum_{k=1}^{K} a_{BBD2_{ik}}\ T_{D2_k} + \left(\alpha - \frac{dz_c}{dx}\right)_i \cos\theta_i$$

where $(\alpha - \frac{dz_c}{dx})$ is the local angle-of-attack, and

$$a_{BBD1_{ik}} = \frac{\cos\theta_i}{2r_i^2}\left\{\beta^2 r_i^2 \cosh^{-1}\frac{x_i - x_k}{\beta r_i}\right.$$

$$\left. + (x_i - x_k)\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}\right\}$$

$$a_{BBD2_{ik}} = -\frac{\cos\theta_i}{r_i^2}\left\{\left[\frac{(x_i - x_k)^2 - 4\beta^2 r_i^2}{3}\right]\right.$$

$$\left.\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2} + (x_i - x_k)\,\beta^2 r_i^2 \cosh^{-1}\frac{x_i - x_k}{\beta r_i}\right\}$$

$$u_{BBD1_{ik}} = \frac{\cos\theta_i}{r_i}\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}$$

$$u_{BBD2_{ik}} = \frac{\cos\theta_i}{r_i}\left\{(x_i - x_k)\sqrt{(x_i - x_k)^2 - \beta^2 r_i^2}\right.$$

$$\left. - \beta^2 r_i^2 \cosh^{-1}\frac{x_i - x_k}{\beta r_i}\right\}$$

Since both $u_{BBD_{ik}}$ and $a_{BBD_{ik}}$ are functions of $\cos\theta_i$, these two terms are redefined

$$u'_{BBD_{ik}} = u_{BBD_{ik}}/\cos\theta_i$$

$$a'_{BBD_{ik}} = a_{BBD_{ik}}/\cos\theta_i$$

and

$$\left(\alpha - \frac{dz_c}{dx}\right)_i = -\left[a'_{BBD1_{i1}} - \left(\frac{dr}{dx}\right)_i u'_{BBD1_{i1}}\right] T_{D1} -$$

$$\sum_{k=1}^{K} \left[a'_{BBD2_{ik}} - \left(\frac{dr}{dx}\right)_i u'_{BBD2_{ik}}\right] T_{D2_k}$$

Again, a linearly varying doublet is placed at the nose of the body, the strength of which is given by:

$$T_{D1} = \frac{-\left(\alpha - \frac{dz_c}{dx}\right)_{nose}}{\frac{\beta}{2}\left[\sqrt{\cot^2\delta - \beta^2} + \beta\cosh^{-1}\left(\frac{\cot\delta}{\beta}\right) + \tan\delta\left(\sqrt{\cot^2\delta - \beta^2}\right)\right.}$$

A quadratically varying doublet is also placed at the nose

$$T_{D2_1} = \frac{-\left(\alpha - \frac{dz_c}{dx}\right)_1 + \left[a'_{BBD1_{11}} - \left(\frac{dr}{dx}\right)_1 u'_{BBD1_{11}}\right] T_{D1}}{a'_{BBD2_{11}} - \left(\frac{dr}{dx}\right)_1 u'_{BBD2_{11}}}$$

and at all subsequent body stations

$$T_{D2_i} = \frac{-\left(\alpha - \frac{dz_c}{dx}\right)_i + \left[a'_{BBD1_{i1}} - \left(\frac{dr}{dx}\right)_i u'_{BBD1_{i1}}\right] T_{D1} + \sum_{k=1}^{i-1} \left[a'_{BBD2_{ik}} - \left(\frac{dr}{dx}\right)_i u'_{BBD2_{ik}}\right] T_{D2_i}}{a'_{BBD2_{ii}} - \left(\frac{dr}{dx}\right)_i u'_{BBD2_{ii}}}$$

At the end of a closing body, a quadratically varying doublet is defined:

$$T_{D2_{NBODYS}} = \frac{-\left(x_{NBODYS} - x_1\right) T_{D1} - \sum_{k=1}^{NBODYS-1} \left(x_{NBODYS} - x_k\right)^2 T_{D2_k}}{\left[x_{NBODYS} - (x - \beta\, r)_{NBODYS_{-1}}\right]^2}$$

The pressure coefficients on the body due to the line sources and doublets are calculated using one of three formulas (as specified by an input).

For the linear pressure coefficient,

$$C_p = -2u_{B_i};$$

for the nonlinear pressure coefficient,

$$C_p = -2u_{B_i} + \beta^2 u_{B_i}^2 - v_{r_{B_i}}^2 - v_{\theta_{B_i}}^2$$

and for the "exact" isentropic pressure coefficient,

$$C_p = \frac{2}{\gamma M^2}\left\{\left[1 - \frac{\gamma - 1}{2} M^2 \left(2u_{B_i} + u_{B_i}^2 + v_{r_{B_i}}^2 + v_{\theta_{B_i}}^2\right)\right]^{\frac{\gamma}{\gamma - 1}} - 1\right\}$$

where

$$u_{B_i} = u_{BBS1_{i1}} T_{S1} + \sum_{k=1}^{K} u_{BBS2_{ik}} T_{S_k} + u_{BBD1_{i1}} T_{D1} +$$

$$\sum_{k=1}^{K} u_{BBD2_{ik}} T_{D_k}$$

$$v_{r_{B_i}} = v_{r_{BBS1_{i1}}} T_{S1} + \sum_{k=1}^{K} v_{r_{BBS2_{i1}}} T_{S2_k} + v_{r_{BBD1_{i1}}} T_{D1} +$$

$$\sum_{k=1}^{K} v_{r_{BBD2_{ik}}} T_{D2_k}$$

$$v_{\theta_{B_i}} = v_{\theta_{BBS1_{i1}}} T_{S1} + \sum_{k=1}^{K} v_{\theta_{BBS2_{i1}}} T_{S2_k} + v_{\theta_{BBD1_{i1}}} T_{D1} +$$

$$\sum_{k=1}^{K} v_{\theta_{BBD2_{ik}}} T_{D_k}$$

and

$$\beta = M^2 - 1$$

$$\gamma = 1.4$$

The lift, drag, and pitching moments on the body due to the line sources and doublets are calculated neglecting any interference effects from the wing. Such interference terms are added later. These forces are determined by integrating the pressures over the body. It is convenient to resolve the aerodynamic forces acting on a body into an axial force X, a normal force N, and a pitching moment M about the nose. The corresponding dimensionless coefficients are given by:

$$C_{X_B} = \frac{X}{qS} = \frac{1}{S}\int_0^1 r \cdot \frac{dr}{dx} \int_0^{2\pi} C_p \, d\theta dx \;;$$

$$C_{N_B} = \frac{N}{qS} = \frac{1}{S} \int_0^1 r \int_0^{2\pi} C_p \cos\theta d\theta dx;$$

$$C_{M_{B_o}} = \frac{M}{qS\bar{c}} = \frac{1}{S\bar{c}} \int_0^1 x \cdot r \int_0^{2\pi} C_p \cos\theta d\theta dx.$$

These coefficients are evaluated in subroutine COEFS using a numerical integration technique.

The total lift, drag, and pitching moments about an arbitrary point are obtained by an appropriate resolution of forces:

$$C_{L_B} = C_{N_B} \cos\alpha - C_{X_B} \sin\alpha$$

$$C_{D_B} = C_{N_B} \sin\alpha + C_{X_B} \cos\alpha$$

$$C_{M_B} = C_{M_{B_o}} + \bar{x}\, C_{N_B} - \bar{z}\, C_{X_B},$$

where the moments are computed about the point $(\bar{x}, \bar{z})$, $\bar{c}$ is the reference chord length, and S is the reference area. The forces and moments are computed for the half-body only.

USAGE:

CALL KARMOR (NBODYS, NTHETA, XN, YN, ZN, CPCALC, VOUT, ARA, XP, ZP, RFAREA, XB, R, THETAB, ZDELTA, XC, YC, ZC, THETA, U, VV, WW, AN1, CPB, BBCL, BBCD, BBCM)

COMMON DATE (2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION XB (NBODYS), R (NBODYS), THETAB (NTHETA), ZDELTA (NBODYS), XC (NBODY + NWING), YC (NBODY + NWING), ZC (NBODY + NWING), THETA (NTHETA), U (NWING), VV (NWING), WW (NWING), AN1 (NWING), CPB (NBODYS, NTHETA)

Input: NBODYS = Number of body stations.
NTHETA = Number of body meridian lines (or $\theta$'s).
XN, YN, ZN = x-, y- and z-coordinates of body nose.
CPCALC = Code.
= 0., calculation of pressure coefficients to use linear equation.
= 1., calculation of pressure coefficients to use nonlinear equation.
= 2., calculation of pressure coefficients to use "exact" isentropic equation.
VOUT = Code.
= 0., velocity components are not to be printed.
= 1., velocity components are to be printed.
ARA = Angle-of-attack of body with respect to freestream.
XP, ZP = x- and z-coordinates of point about which pitching moments are to be computed.
RFAREA = Reference area.
XB = Array of x-coordinates of body stations.
R = Array of body radii at body stations.
THETAB = Array of $\theta$-angles of body meridian lines.
ZDELTA = Array of body cambers at body station
XC = Array of panel control point x-coordinates.
YC = Array of panel control point y-coordinates.
ZC = Array of panel control point z-coordinates.
THETA = Scratch array.
NBODY = Number of body panels.
NWING = Number of wing panels.
XMACH = Mach number.
U = Array of x-components of velocity on the wing panels due to the body sources and doublets.
VV = Array of y-components of velocity on the wing panels due to the body sources and doublets.
WW = Array of z-components of velocity on the wing panels due to the body sources and doublets.
AN1 = Array of normal components of velocity on the wing panels due to the body sources and doublets.
CPB = Array of pressure coefficients.
BBCL = Lift of the body.

| | | |
|---|---|---|
| | BBCD | = Drag of the body. |
| | BBCM | = Pitching moment of the body. |
| SUBPROGRAMS CALLED: | ALOG<br>SQRT<br>COS<br>SIN<br>ATN1 | = (Built-in functions) |
| | FSF | |
| | DERIVS | |
| | COEFS | |
| ERROR RETURNS: | None | |
| RESTRICTIONS: | NBODYS | $\leq$ 55 |
| | NBODY | $\leq$ 100 |
| | NWING | $\leq$ 110 |
| STORAGE: | $5163_{10}$ | $= 12053_8$ |

SUBJECT: FORTRAN IV Subroutine LACKEY

PURPOSE: Computes perturbation velocities and pressure coefficients at field points due to sources and vortices on the wings and due to vortices on the bodies. When required, the freestream velocity components are added to the results to yield the total velocity at the field point. The user selects the pressure coefficient calculation equations from among linear, nonlinear, or "exact" isentropic.

METHOD: This subroutine calls EVAL1, and VEL1 to compute the peturbation velocities. The pressure coefficient is then computed from the appropriate equation.

Linear: $C_p = -2u$

Nonlinear: $C_p = -2u + \beta^2 u^2 - v^2 - w^2$

Exact Isentropic: $C_p = \frac{1.42857}{M^2} \left\{ \left[ 1 + 2 M^2 (1 - Q^2) \right]^{\frac{7}{2}} - 1 \right\}$

where

u = velocity component in the x-direction.

v = velocity component in the y-direction.

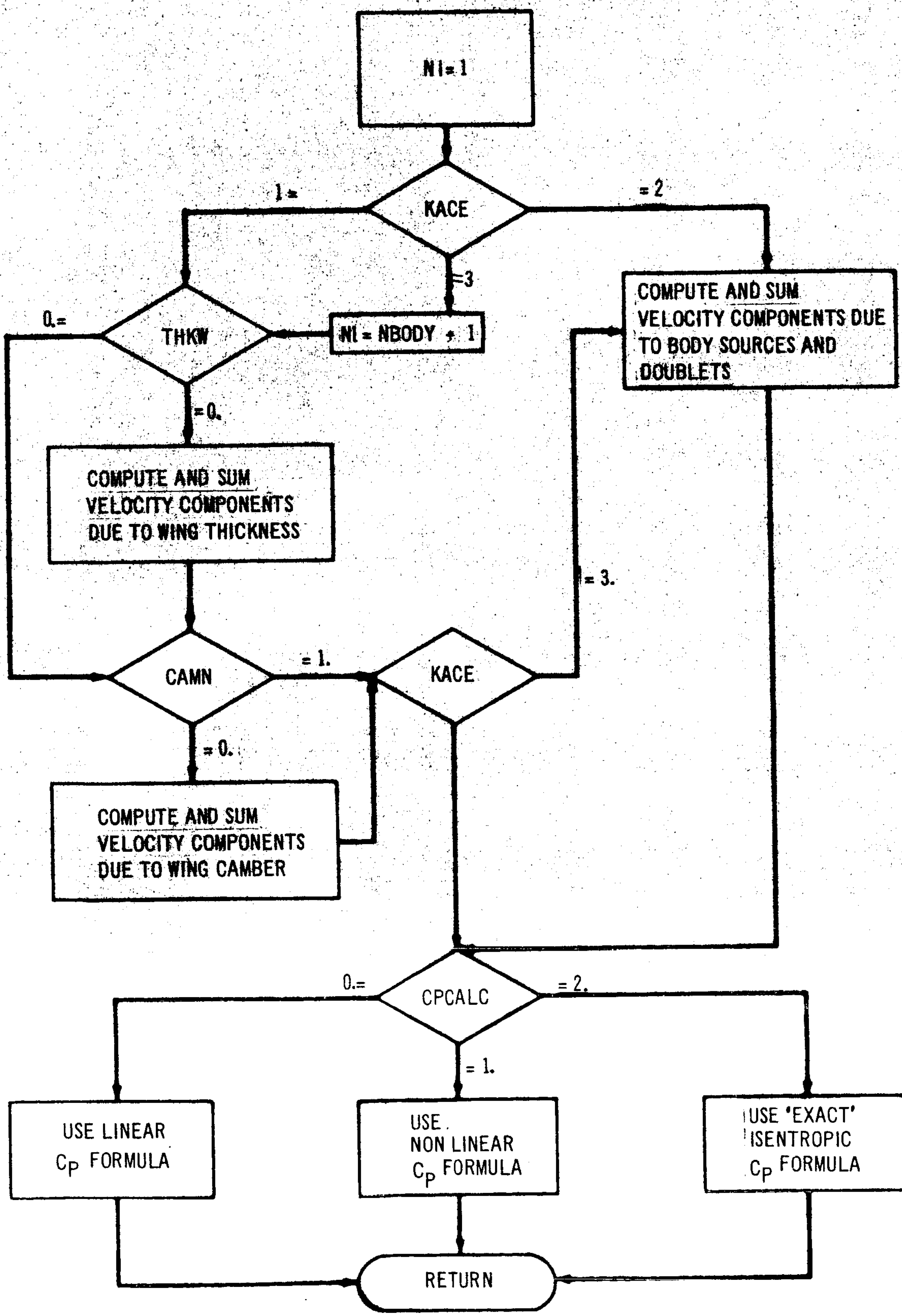
w = velocity component in the z-direction.

M = Mach number.

$\alpha$ = angle of attack.

$Q^2 = (u + \cos\alpha)^2 + v^2 + (w + \sin\alpha)^2.$

A flow chart of the routine follows.


NI = 1
KACE
1 =
= 2
= 3
COMPUTE AND SUM VELOCITY COMPONENTS DUE TO BODY SOURCES AND DOUBLETS
THKW
0.=
NI = NBODY + 1
= 0.
COMPUTE AND SUM VELOCITY COMPONENTS DUE TO WING THICKNESS
= 3.
CAMN
= 1.
KACE
= 0.
COMPUTE AND SUM VELOCITY COMPONENTS DUE TO WING CAMBER
CPCALC
0.=
= 2.
= 1.
USE LINEAR $C_P$ FORMULA
USE NON LINEAR $C_P$ FORMULA
USE 'EXACT' ISENTROPIC $C_P$ FORMULA
RETURN

USAGE:

COMMON/THICK/THKW, ARA, CPCALC, CAMN, POP

COMMON/FLOV1/KACE, NBODY, DUM (5), XMACH

COMMON/FLOV5/N1, N2, N3, N4

CALL LACKEY (ISTR, XB, YB, ZB, U, V, W, CP)

| Input: | | |
|---|---|---|
| | THKW | = 1. indicates a thick wing. |
| | | = 0. camber only. |
| | ARA | = angle of attack in radians. |
| | CPCALC | = 0. linear equations used for C. |
| | | 1. nonlinear equations used for C. |
| | | 2. "exact" isentropic equations used for C. |
| | CAMN | = 0. indicates velocity components due to camber to be computed. |
| | | = 1. indicates velocity components due to camber not to be computed. |
| | KACE | = 1 wing alone. |
| | | = 2 body alone. |
| | | = 3 wing and body. |
| | NBODY | = number of vortex panels on body. |
| | XMACH | = Mach number. |
| | XB | = x-coordinate of field point. |
| | YB | = y-coordinate of field point. |
| | ZB | = z-coordinate of field point. |
| | ISTR | = 1. indicates freestream velocity to be added to velocity components. |
| | | = 0. indicates freestream velocity not to be added at field point. |

| | | |
|---|---|---|
| | N1, N2, N3, N4 | = 4 logical tape numbers on which velocity components due to the various singularities are written for use by FLOOUT. |
| Output: | U | = x-direction velocity component. |
| | V | = y-direction velocity component. |
| | W | = z-direction velocity component. |
| | CP | = $C_p$ at field point. |

SUBPROGRAMS CALLED: EVAL1
VEL1

ERROR RETURN: None

RESTRICTIONS: None

STORAGE: $428_{10} = 654_8$

| | |
|---|---|
| SUBJECT: | MAP Function LOC |
| PURPOSE: | To determine the absolute machine address of a variable or first location of an array. This routine is used only in the streamline integration package by DEDIS. |
| METHOD: | Using machine hardware instructions and standard system conventions, the address of the argument is returned in the AC to the calling program. |
| USAGE: | INTEGER VARADD<br><br>VARADD = LOC (VAR)<br><br>Input: VAR = any array or variable name.<br><br>Output: VARADD = contains the address of VAR. |
| SUBPROGRAMS CALLED: | None |
| RESTRICTIONS: | None |
| STORAGE: | $4_{10} = 4_8$ |

SUBJECT: FORTRAN IV Subroutine MDMATE

PURPOSE: To compute the matrix resulting from the application of the Lagrange multipliers technique in minimizing the drag for given wing constraints.

METHOD: The minimum drag of a wing in the presence of a body for given wing $C_L$ and pitching moment may be determined by applying the method of Lagrange multipliers to the system of equations defining the drag. A function F is defined in terms of the pressure difference of the wing panels, $p_{W_i}$, and the auxiliary variables (Lagrange multipliers) $\lambda_1$ and $\lambda_2$.

$$F = D + \lambda_1 (L - \overline{L}) + \lambda_2 (M - \overline{M})$$

The condition for minimum drag for the constraints that L = L and M = M is found by setting the partial derivatives of F with respect to the variable $p_{W_i}$, $\lambda 1$, and $\lambda 2$ equal to zero. The result is the following system of linear equations, and the pressure difference across the wing panels may now be determined for minimum drag.

$$\begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & & & -A_2 & \cdot \\ \vdots & & & \vdots & \vdots \\ \cdot & & & -A_N & -x_N A_N \\ -A_1 & & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \overline{L} \\ \overline{M} \end{Bmatrix}$$

where $\bar{n}_i$ = The downwash on the wing due to the body sources.

$\bar{n}_{B_i}$ = The downwash on the wing resulting from the cancelation of normal velocity on the body due to the wing thickness.

If only the lift is constrained, the row and column of the matrix corresponding to $\lambda_2$ are omitted and $p_{W_i}$ is found by inverting as before.

In this subroutine, the coefficients of the system of equations are computed and the inverses for the two cases are stored on a scratch tape in preparation for wing optimization by a later link. The matrices are stored with an end-of-file mark separating them.

USAGE:

CALL MDMATE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION WW(115, 115), XBAR(210), AREA(210)

| | | | |
|---|---|---|---|
| Input: | XBAR | = | Array of panel centroid x-coordinates. |
| | AREA | = | Array of panel areas. |
| | NBODY | = | Number of body panels. |
| | NWING | = | Number of wing panels. |
| | WW | = | Reduced matrix of the aerodynamic influence coefficients. |
| | NTAPEE | = | Logical tape number from which the reduced matrix of the aerodynamic influence coefficients is read. |
| | NTAPEA | = | Logical tape number on which the drag minimization matrices is stored. |
| Output: | WW | = | Drag minimization matrix written on NTAPEA. |
| | $WW^{-1}$ | = | Inverse of the drag minimization matrix constrained for wing lift written on NTAPEA. |
| | $WW^{-1}$ | = | Inverse of the drag minimization matrix constrained for wing lift and pitching moment written on NTAPEA. |

SUBPROGRAM CALLED:

SINVRT (see Appendix B)

ERROR RETURNS:

If an error occurs in the inversion of the matrix, a message will be written on the output tape "ERROR IN INVERSION OF DRAG MINIMIZATION MATRIX" followed by the error codes from subroutine SINVRT.

RESTRICTIONS: The matrix to be inverted must not exceed the order 112.

STORAGE: $14218_{10} = 33612_8$

SUBJECT: FORTRAN IV Subroutine MEAN

PURPOSE: To smooth a set of data points defining a single-valued function of one variable.

METHOD: A sliding operation that deals with four points at a time $\left((x_i, y_i), i = 1, 2, 3, 4\right)$ is employed. Points whose ordinates may be changed are the interior two points at each setting of the operation. The abscissae of the input points are not changed. A straight line is drawn from $(x_1, y_1)$ to $(x_3, y_3)$ and another straight line from $(x_2, y_2)$ to $(x_4, y_4)$. If the abscissa of the intersection of these two lines lies in the interval $(x_2, x_3)$, then the points are left as is, and the operation slides over one point and begins anew. If the lines are parallel or if the abscissa of the intersection is outside $(x_2, x_3)$, then new ordinates $y_2$ and $y_3$ are calculated. An ordinate on the first line is evaluated at the abscissa $x_2$; similarly an ordinate on the second line is evaluated at the abscissa $x_3$. The new values $y_2$ and $y_3$ are the mean values of the original ordinates and new ordinates so calculated.

This operation slides along the set of points until all interior points have been subjected to smoothing. Five such passes made over the curve at each call of MEAN.

USAGE: CALL MEAN (X, Y, N)

Input:
- X = Array of abscissae; a strictly monotone sequence
- Y = Array of ordinates to be smoothed
- N = Number of points

Output:
- Y = Smoothed ordinates (the first and last points are not subjected to smoothing)

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: The abscissae must be strictly monotonic; thus vertical slopes are excluded.

STORAGE: $163_{10} = 243_8$

SUBJECT: FORTRAN IV Function MERR

PURPOSE: To write error messages for subroutines in the geometric definition portion of the program.

METHOD: An error array NU is given. NU(1) is an error code that should be the same as a given success code IOK if an error condition does not exist. NU(2), if greater than zero, is the number of an output tape on which an error message can be written. NU(3) is an error message limiter; a message is not to be written unless NU(3) is greater than 1. M is an error message number.

If NU(1) = IOK, the function value (K) is set to zero, and no further action is necessary. Otherwise, NU(3) is decreased by 1; then if NU(2) $\leq 0$ or if NU(3) $\leq 0$, K is set to -1 and return is made to the calling program. Otherwise, the following message is written:

"ERROR i, CODE j, IN SUBROUTINE xxxxxx DURING GEOMETRIC DEFINITION"

where i = M, j = NU(1), and xxxxxx is the given name of the subroutine in which the error occurred.

USAGE: DIMENSION NU(3)

K = MERR (NU, IOK, M, SR)

Input: NU = Error array.

IOK = Success code.

M = Error message number.

SR = Subroutine name (alphameric).

Output: Message on error tape.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $87_{10} = 127_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Function NWEED |
| PURPOSE: | To remove excess points from an array of points in m-space. |
| METHOD: | Each point is compared with the previous point. If all corresponding coordinates are the same with given tolerances, that point is eliminated by shifting the balance of the array and reducing the number of points by 1. However, if the last point is too near its neighbor, the neighbor is discarded and the last point retained. |
| USAGE: | DIMENSION A(M, N), EP(M)<br>NN = NWEED (A, N, M, EP) |

| | | |
|---|---|---|
| Input: | A | = Given array of points. |
| | N | = Number of points in A. |
| | M | = Space dimension. |
| | EP | = Array of tolerances, one for each coordinate. |
| Output: | A | = Weeded array of points. |
| | NN | = Number of points in the weeded array (NN may have the same storage location as N). |

| | |
|---|---|
| SUBPROGRAM CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | $N > 1$ and $M > 0$. If any $EP(i) \leq 0$, no weeding will be done. For weeding only on selected coordinates, EP is set to some large value for the remaining coordinates. |
| STORAGE: | $153_{10} = 231_8$ |

SUBJECT: FORTRAN IV Program OPCAM

PURPOSE: To interpret data commands and control flow through the Program for Analysis and Design of Supersonic Wing-Body Combinations.

METHOD: The program first calls subroutine OPCAMI, which carries out the following operations: (1) assigns logical tape numbers to tape symbols (in COMMON); (2) obtains the data and places it in COMMON; (3) writes the program title on the output tape; (4) transfers the data cards to a scratch tape, which is then used as the input tape for the remainder of the program; (5) writes the data cards on the output tape.

Then function INTURP is used to read a data card, to write its contents on the output tape, and to see whether it is a command card. A command card has the words DEFINE, DEF1NE, PANEL, AERODY, AREODY, END OF or END 0F in columns 1-6; the contents of the remaining columns are written on the output tape but are not otherwise used. If the data card is a command card, the appropriate action is taken; otherwise, successive cards are read until a command card is found (this feature permits the insertion of comment cards or blank cards immediately before any command card).

The command DEFINE (or DEF1NE) results in a call to subroutine GEOMD. On return, data cards are again read until a command card is encountered. If any GEOMD data cards were not read because of error, they will be listed by INTURP.

Subroutine TFLAT is called when a PANEL card is read. If TFLAT detects an error, it skips over data cards until the input tape is positioned just in front of a DEFINE or END OF card; OPCAM then goes on to process this card in the usual way. But if TFLAT signals success, then the argument NOBODY (set by TFLAT) is examined. If NOBODY = -1, meaning a body-alone case, OPCAM goes on to read the next data card; otherwise, OPCAM calls subroutine PANEL.

OPCAM calles subroutine AERO when an AERODY (or AER0DY) card is read. An END OF (or END 0F) card results in program termination.

| | |
|---|---|
| SUBPROGRAMS CALLED: | AERO<br>CLOK<br>GEOMD<br>INTURP<br>TFLAT<br>PANEL<br>OPCAM1 |
| ERROR RETURNS: | Program errors are dealt with in the program section where they occur. |
| RESTRICTIONS: | Each time that INTURP reads and writes a data card, OPCAM writes (on the same line, near the right margin) the word TIME followed by the time of day. The time of day is found by a subroutine CLOK. The basic program deck contains a dummy subroutine CLOK which merely returns blanks. A proper subroutine CLOK may replace the dummy at the installation where the program is used. |
| STORAGE: | $152_{10} = 230_8$ |

SUBJECT: FORTRAN IV Subroutine OPCAMI

PURPOSE: To perform initialization chores for the main program OPCAM.

METHOD: Tape symbols are assigned as follows:

| Symbol | Tape Number | Description |
|---|---|---|
| LIN | 5 | System input tape. |
| LO | 6 | System output tape. |
| LI | 9 | Input tape used by remainder of program. |
| LA | 1 | Scratch tapes. |
| LB | 2 | |
| LC | 3 | |
| LD | 4 | |
| LE | 7 | |
| LF | 8 | |

Subroutine DATE is called to obtain the date. A title is written on the output tape. Tape LI is rewound. Then all data cards are read with FORMAT(13A6, A2) from tape LIN, written on tape LO with FORMAT(1H 14A6), and written on tape LI with FORMAT(13A6, A2); the last data card is signaled by the word END OF or END 0F (in case the letter O is mistakenly keypunched as the number zero) in columns 1-6. Then an EOF is written on tape LI, and it is rewound.

USAGE:

```
COMMON  DAT(2), LA, LB, LC, LD, LE, LF, LI, LO
CALL OPCAMI
```

Output: DAT = Date (alphameric)

LA, LB, LC, LD, LE — Scratch tape numbers.

LF = Scratch tape number.

LI = Input tape number.

LO = Output tape number.

SUBPROGRAM CALLED: DATE

ERROR RETURNS: None

RESTRICTIONS: An END OF card must be placed at the end of the data deck. If the END OF card does not appear in the data deck, the job will be terminated by the EOF on the system input tape. If the END OF card is not the physically last card, only the data preceding it will be read by the remainder of the program.

STORAGE: $235_{10} = 353_8$

SUBJECT: FORTRAN IV Subroutine OPTIM3

PURPOSE: Given four successive points on the graph of a function of a single variable, to generate additional points between one selected pair of the given points, the spacing of the additional points to be such that straight-line segments joining them will represent the given function with a prescribed maximum error.

METHOD Four successive arguments $x_i$ and their respective ordinates $y_i$ are input to the subroutine, together with a tolerance z and integer k to indicate the interval ($x_k$, $x_{k+1}$) to consider. OPTIM3 then generates the biquadratic interpolating function described in subroutine BITURP. This function is a linear combination of two quadratics on the interval ($x_2$, $x_3$), hence it is there a cubic; on either of the other possible intervals it is a quadratic. In any case, the function has a continuous first derivative, unlike a Lagrangian interpolation polynomial of any order.

Having generated the interpolation function $\varphi$, OPTIM3 computes its function value y, its slope S, and its radius of curvature R, all at some particular x (at $x_k$ in the first instance). In a small neighborhood of (x, y) the function $\varphi$ can be represented by a circular arc of radius R, tangent at (x, y) to $\varphi$. OPTIM3 then computes the central angle of that arc so that the arc and its chord have a maximum separation, in the y-direction, of z. Then the x-component $\Delta x$ of that chord is the distance from the current argument x to the next argument (unless $x_{k+1}$ has been reached), and the process continues.

USAGE: DIMENSION X(4), Y(4,) A(200), B(200)

CALL OPTIM3 (X, Y, Z, A, B, K, L, M)

Input: X = Array of x-coordinates of given points.

Y = Array of y-coordinates of given points.

Z = Allowable tolerance (in y direction).

K = Subscript of x at lower end of interval to be interpolated.

Output: A = Array of x-coordinates of interpolated points. A(1) = X(K), A(2) = x of first interpolated point, etc.

B = Array of y-coordinates of interpolated points. B(1) = Y(K), B(2) = y of first interpolated point, etc.

L = One more than the number of interpolated points generated by the routine.

M = Error indicator, which is 1 if success.

SUBPROGRAMS CALLED: ACOS, ATAN, COS (Built-in functions), SQRT

ERROR RETURNS: M = 0 if X(1) = X(4); M = 2 if L = 200 and more points are needed.

RESTRICTIONS: The $x_i$ must form a strictly monotonic sequence, though not necessarily equally spaced. Note that the A and B arrays contain the first given point, but not the last.

STORAGE: $594_{10} = 1122_8$

SUBJECT: FORTRAN IV Subroutine OPTMW

PURPOSE: For wing alone, to compute the pressure difference across wing panels for a wing optimized for minimum drag.

METHOD: The method is outlined in the description of subroutine MDMATE. From the matrix inverse computed in MDMATE, the following multiplication yields the pressure difference across the panels.

$$\begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{bmatrix} -2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \ldots & & -A_1 & -x_1 A_1 \\ -(A_2 a_{21} + A_1 a_{12}) & \cdot & & -A_2 & \cdot \\ \vdots & & & \vdots & \vdots \\ \cdot & & & -A_N & -x_N A_N \\ -A_1 & \cdot & \cdot\ -A_N & 0 & 0 \\ -x_1 A_1 & \cdot & \cdot\ -x_1 A_N & 0 & 0 \end{bmatrix}^{-1} \cdot \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \bar{L} \\ \bar{M} \end{Bmatrix}$$

If only the lift is constrained, the row and column of the matrix corresponding to $\lambda_2$ is omitted and $p_{W_i}$ is found as before.

USAGE: CALL OPTMW(NW, NTAPEX, A, B, CONSNT, CLBAR, XCPBAR, RFAREA, AREA, CL)

DIMENSION A(1), B(1), AREA(1), CL(1)

Input:

| | | |
|---|---|---|
| NW | = | Number of wing panels. |
| NTAPEX | = | Logical number of tape from which the inverse of the minimum drag matrix is read. |
| A | = | Dummy array used by the subroutine. |
| B | = | Dummy array used by the subroutine. |
| CONSNT | = | 0., indicates wing optimization for wing $C_L$ constraint. |
| | = | 1., indicates wing optimization for wing $C_L$ and pitching moment constraints. |
| CLBAR | = | Wing coefficient of lift constraint. |

| | | | |
|---|---|---|---|
| | | XCPBAR | = x-coordinate of the wing center of pressure constraint. |
| | | RFAREA | = Wing reference area. |
| | | AREA | = Array of wing panel areas. |
| | Output: | CL | = Array of the pressure difference across the wing panels optimized for minimum drag. |

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $131_{10} = 203_8$

SUBJECT: FORTRAN IV Subroutine OPTMWB

PURPOSE: In the presence of a body, to compute the pressure difference across wing panels for a wing optimized for minimum drag.

METHOD: The method is outlined in the descritpion of subroutine MDMATE. From the matrix inverse computed in MDMATE the following multiplication is carried out yielding the pressure difference across the panels.

$$
\begin{Bmatrix} p_{W_1} \\ p_{W_2} \\ \vdots \\ p_{W_N} \\ \lambda_1 \\ \lambda_2 \end{Bmatrix}
=
\begin{bmatrix}
-2A_1 a_{11} & -(A_1 a_{12} + A_2 a_{21}) \cdots & & -A_1 & -x_1 A_1 \\
-(A_2 a_{21} + A_1 a_{12}) & & & -A_2 & \cdot \\
\vdots & & & \vdots & \vdots \\
\cdot & & & -A_N & x_N A_N \\
-A_1 & & \cdot \; -A_N & 0 & 0 \\
-x_1 A_1 & & \cdot \; -x_1 A_N & 0 & 0
\end{bmatrix}^{-1}
\cdot
\begin{Bmatrix} A_1(\bar{n}_1 + \bar{n}_{B_1}) \\ A_2(\bar{n}_2 + \bar{n}_{B_2}) \\ \vdots \\ A_N(\bar{n}_N + \bar{n}_{B_N}) \\ \bar{L} \\ \bar{M} \end{Bmatrix}
$$

where $\bar{n}_i$ = The downwash on the wing due to body sources

$$\bar{n}_{B_i} = \sum_{j=1}^{NBODY} D_{ij} \cdot (-n_{B_j})$$ is the downwash on the wing due to the cancellation of the normal velocity on the body due to wing thickness.

$D_{ij}$ = The interference influence coefficients matrix.

$n_{Bj}$ = Downwash on the body due to wing sources

$\bar{L} = C_L \cdot S_W$ is the lift constraint

$\bar{M} = \bar{X} \cdot \bar{C}_L \cdot S_W$ is the pitching moment constraint

If only the lift is constrained, the row and column of the matrix corresponding to $\lambda_2$ is omitted and $p_{W_i}$ is found as before.

CALL OPTMWB(NW, NB, NTAPEX, NTAPEY, THICK, A, B, ALPHA, ABX, ALPHAX, AREA, CONSNT, CLBAR, XCPBAR, RFAREA, CL)

DIMENSION A(1), B(1), ALPHA(1), ABX(1), ALPHAX(1), AREA(1), CL(1)

Input:

| | | | |
|---|---|---|---|
| Input: | NW | = | Number of wing panels. |
| | NB | = | Number of body panels. |
| | NTAPEX | = | Logical number of the tape from which the inverse of the minimum drag matrix is read. |
| | NTAPEY | = | Logical number of the tape from which D is read. |
| | THICK | = | 0., wing thickness effects not to be included. |
| | | = | 1., wing thickness effects to be included. |
| | A | = | Dummy array used by the subroutine. |
| | B | = | Dummy array used by the subroutine. |
| | ALPHA | = | Dummy array used by the subroutine. |
| | ABX | = | Array of normal on the body due to a given wing thickness distribution. |
| | ALPHAX | = | Array of downwash values on the wing due to body sources. |
| | AREA | = | Array of panel areas. |
| | CONSNT | = | 0., wing optimization for wing $C_L$ constraint |
| | | = | 1., wing optimization for wing $C_L$ and pitching moment constraints. |
| | CLBAR | = | Wing coefficient of lift constraint. |
| | XCPBAR | = | x-coordinate of the wing center of pressure constraint. |
| | RFAREA | = | Wing reference area. |
| Output: | CL | = | Array of the pressure difference across the wing panels optimized for minimum drag. |

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $239_{10} = 357_8$

SUBJECT: FORTRAN IV Subroutine OUTB

PURPOSE: To print out the data for body panels in a prescribed format.

METHOD: The data for the region of the body represented by panels are printed in the following format:

| THETA (DEG.) ROW NO. | $\theta_1$ | $\theta_2$ | . . . | $\theta_{NTHETA}$ |
|---|---|---|---|---|
| 1 | $z_{1,1}$ | $z_{1,2}$ | . . . | $z_{1,NTHETA}$ |
| 2 | $z_{2,1}$ | | | . |
| 3 | . | | | . |
| . | . | | | . |
| . | . | | | . |
| . | . | | | . |
| NROW | $z_{NROW,1}$ | . | . . . | $z_{NROW,NTHETA}$ |

The numbers running horizontally represent the angles, $\theta$ around the body starting at the crown line and proceeding to the keel line. The numbers running vertically represent the rows of panels starting from the furthest forward panel and working aft.

USAGE:

```
CALL OUTB (NTAPEO, NBODY, NTHETA, NROWB,
           THETA, Z)
DIMENSION  Z(NBODY), THETA(NTHETA)
```

Input:
- NTAPEO = Logical number of output tape.
- NBODY = Number of body panels.
- NTHETA = Number of $\theta$ angles.
- NROWB = Number of rows of panels.
- THETA = $\theta$ angles.
- Z = Data to be output.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $119_{10} = 167_8$

SUBJECT: FORTRAN IV Subroutine OUTPTB

PURPOSE: To write body paneling data on output tape as printout and on scratch tape for use in other program sections.

METHOD: Program uses direct write tape statements to write body paneling data on output tape and on scratch tape (see Appendix D for tape format). Program uses a tolerance of ±0.00001 and sets all small negative and positive values equal to 0. Program also checks and eliminates any trivial secondary panel parts. An input tolerance is used for this check.

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NTAPE2
NWRITE

Output: NBODY

COMMON /COM1/ (See subroutine PANEL)

Input: NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
XCEN
YCEN
ZCEN
XCON
YCON
ZCON
AREA
THETA
ALPHA
CHORD

CALL OUTPTB

| | |
|---|---|
| SUBPROGRAM CALLED: | SQRT (Built-in function) |
| ERROR RETURNS: | None |
| STORAGE: | $1392_{10} = 2560_8$ |

SUBJECT: FORTRAN IV Subroutine OUTPTW

PURPOSE: To write wing paneling data on output tape as printout and on scratch tape for use in other program sections.

METHOD: Program uses direct write tape statements to write wing paneling data on output tape and on scratch tape (see Appendix D for tape format). Program uses a tolerance of ±0.00001 and sets all small negative and positive values equal to 0. Program also checks all secondary panel parts of two-part panels on inboard and outboard columns and eliminates any "small" parts. An input tolerance is used for this check.

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input: NTAPE2
NWRITE

Output: NWING

COMMON /COM1/ (See subroutine PANEL)

Input: KSTART
KEND
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: KPANEL
XCOR
YCOR
ZCOR
XINT
YINT
ZINT
XCEN
YCEN
ZCEN
XCON
YCON
ZCON
AREA
THETA
ALPHA
CHORD

CALL OUTPTW

SUBJECT: FORTRAN IV Subroutine OUTW

PURPOSE: To print out the data for wing panels in a prescribed format.

METHOD: The data for the wing panels are output in the following format:

| SPANWISE STATION<br>CHORDWISE STATION | 1 | 2 | 3 . . . | NCOL |
|---|---|---|---|---|
| 1 | $Z_{1,1}$ | $Z_{1,2}$ | . . . . | $Z_{1,NCOL}$ |
| 2 | $Z_{2,1}$ | | | |
| 3 | . | | | . |
| . | . | | | . |
| . | . | | | . |
| . | . | | | . |
| NROW | $Z_{NROW,1}$ | | . . . . | $Z_{NROW,NCOL}$ |

The numbers running horizontally represent NCOL columns of panels from inboard to outboard, and the numbers running vertically represent NROW rows of panels from the leading to the trailing edge.

USAGE: CALL OUTW (NTAPEO, NWING, NCOLW, NROWW, Z)

DIMENSION Z(NWING)

Input: NTAPEO = Logical number of output tape.

NWING = Number of wing panels.

NCOLW = Number of chordwise columns of panels.

NROWW = Number of rows of panels in a chordwise column.

Z = Data to be output.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $119_{10} = 167_8$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

STORAGE: $1612_{10} = 3114_8$

SUBJECT: FORTRAN IV Subroutine PANEL

PURPOSE: To serve as control program for geometry paneling subroutines.

METHOD: Program first reads control card used in later subroutines to calculate coordinates of panel control point. Program then reads control card containing literal name and transfers to appropriate subroutine. Either the program transfers to the subroutine BODY or to the subroutine WING, or the program returns to the subroutine OPCAM.

The program contains the following labeled COMMON statement that occurs in all lower level geometry paneling subroutines:

```
COMMON/COM1/KODEB, KODEW, KODEWU, KODEI,
KODEC, XPER, YPER, KOPTB, KOPTW, KOPTF,
NUMS, KTYPE, KSTART, KEND, KINT, XI(16),
YI(16), ZI(16), NPER, NPER1, NPLANE,
NPLN1
```

KODEB = Code for body definition error.

KODEW = Code for wing definition error.

KODEWU = Code for error in wing thickness (or camber) calculations.

KODEI = Code for body-wing intersection definition error.

KODEC = Code for panel control point location. If KODEC = 0, panel control point is located on the streamwise chord through the panel centroid. If KODEC = 1, input value for YPER is used to locate control point.

XPER = Fraction of local streamwise panel chord at which panel control point is located. $0. \leq$ XPER $\leq 1.$

YPER = Fraction of local spanwise panel chord at which panel control point is located. $0. <$ YPER $\leq 1.$
NOTE: If YPER = 0, panel control point is located on the streamwise chord through the panel centroid.

| | | |
|---|---|---|
| KOTPB | = | Code for additional body panel card input. If KOPTB = 1, body panel corner points are input on cards. If KOPTB = 0, otherwise. |
| KOPTW | = | Code for additional wing panel card input. If KOPTW = 1, wing panel corner points are input on cards. If KOPTW = 0, otherwise. |
| KOPTF | = | Code for additional wing panel card input. If KOPTF = 1, actual wing surface coordinates are input on cards. If KOPTF = 0, otherwise. |
| NUMS | = | Number of airfoil sections for which wing surface coordinates are read as additional card input. |
| KTYPE, KSTART, KEND | = | Codes for internal program control. |
| KINT | = | Code for body-wing intersection. If KINT = 1, body-wing intersection has been requested and calculated. If KINT = 0, otherwise. |
| XI, YI, ZI | = | x, y, z coordinates of body-wing intersection. |
| NPER | = | Number of columns of panels plus one. |
| NPER1 | = | Number of columns of panels. |
| NPLANE | = | Number of rows of panels plus one. |
| NPLN1 | = | Number of rows of panels. |

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

Input:
NTAPE1
NTAPE2
ND1
NREAD
NWRITE

CALL PANEL

SUBPROGRAMS CALLED: WING

BODY

INTURP

ERROR RETURNS: Error messages indicate whether error occurred in reading various geometry definitions from scratch tape.

STORAGE: $11208_{10} = 25710_8$

SUBJECT: FORTRAN IV Subroutine PARTV

PURPOSE: To partition velocity components due to the surface distribution of vorticity in preparation for use in later aerodynamic links.

METHOD: The velocity components are read from a scratch tape written by subroutine EVAL in the following form, where a column is written as a logical record.

$$\begin{bmatrix} U_{BB} & U_{BW} \\ V_{BB} & V_{BW} \\ W_{BB} & W_{BW} \\ U_{WB} & U_{WW} \\ V_{WB} & V_{WW} \\ W_{WB} & W_{WW} \end{bmatrix}$$

where the subscripts are given:

BB = Influence on the body due to the body

WB = Influence on the wing due to the body

BW = Influence on the body due to the wing

WW = Influence on the wing due to the wing

The velocity components are reordered to a partitioned form with an end-of-file separating the partitions.

$$\begin{bmatrix} U_{BB} \\ V_{BB} \\ W_{BB} \end{bmatrix} \begin{bmatrix} U_{WB} \\ V_{WB} \\ W_{WB} \end{bmatrix} \begin{bmatrix} U_{BW} \\ V_{BW} \\ W_{BW} \end{bmatrix} \begin{bmatrix} U_{WW} \\ V_{WW} \\ W_{WW} \end{bmatrix}$$

USAGE: CALL PARTV

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION U(110), V(110), W(110), UU(110), VV(110), WW(110)

Input: NTAPEB = Logical tape number from which the velocity components are read.

| | | | |
|---|---|---|---|
| | NTAPED | = | Logical tape number on which the velocity components are written. |
| | NWING | = | Number of wing panels. |
| | NBODY | = | Number of body panels. |
| | U, V, W | = | Velocity components in random order on logical tape NTAPEB. |
| Output: | U, V, W | = | Velocity components in a specified order written on logical tape NTAPED. |

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $0 < \text{NBODY} \leq 100$

$0 < \text{NWING} \leq 110$

STORAGE: $1099_{10} = 2113_8$

SUBJECT: FORTRAN IV Subroutine POLXN

PURPOSE: To find the intersections of an m-dimensional polygonal arc and a plane (or line, if $m = 2$). A polygonal arc is defined as a set of points consecutively connected by straight lines.

METHOD: Consider any two consecutive points in the polygonal arc P, say $P_i$ and $P_{i+1}$. If $P_i$ (or $P_{i+1}$) lies within a given distance EP of the plane, it is said to lie in the plane. If one point lies in the plane, that point is returned as an intersection. If both points $P_i$ and $P_{i+1}$ lie in the plane (i.e., the line segment $\overline{P_i\ P_{i+1}}$ is contained in the plane), only the end points of that line segment are returned as intersections. If $P_i$ and $P_{i+1}$ lie on opposite sides of the plane, the intersection of the line segment $\overline{P_i\ P_{i+1}}$ with the plane is returned. The subroutine stops processing points when a specified maximum number of intersections has been found. The point subcripts associated with any intersection are also returned.

USAGE: DIMENSION C(M+1), P(M, N), Q(M, MAX), LQ(2, MAX)

CALL POLXN (C, P, M, N, EP, MAX, Q, LQ, L)

Input:
- C = Coefficients of the plane $C_1x_1 + C_2x_2 + \ldots + C_Mx_M + C_{M+1} = 0.$
- P = Array of point coordinates $x_{11}, x_{21}, \ldots, x_{M1}, x_{12}, x_{22}, \ldots x_{M2}, \cdot$
- M = Dimension of hyperspace.
- N = Number of points in P.
- EP = Tolerance.
- MAX = Maximum number of intersections to be found.

Output:
- Q = Array of intersection points.
- LQ = Array of point numbers of points in P that are adjacent to points in Q; for instance, if an intersection in Q is between the fourth and fifth points of P, the corresponding elements of LQ are (4, 5); if point i in P lies in the plane (or within distance EP from it), then the point i appears in Q,

and the corresponding elements in LQ are (i, i).

L = Number of intersection points in Q; L = 0 if no intersections are found.

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS: L will be returned as -1 if $M \leq 0$, or if $MAX \leq 0$, or if all of the coefficients C are zero.

RESTRICTIONS: If any points are identical and lie in the plane, each such point will be returned as an intersection.

STORAGE: $268_{10} = 414_{8}$

SUBJECT: FORTRAN IV Subroutine QRAT

PURPOSE: To find the coefficients of the quadratic that passes through three given points, and optionally to evaluate the quadratic at a given value of x.

METHOD: The quadratic equation is

$$y = C_1 + C_2 x + C_3 x^2$$

Let $(x_i, y_i)$, $i = 1, 2, 3$ be the three given points. Then if no two points have the same x-coordinate,

$$C_3 = \left(\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}\right) \Big/ (x_3 - x_1)$$

$$C_2 = (y_3 - y_1) \Big/ (x_3 - x_1) - C_3 (x_3 + x_1)$$

$$C_1 = y_2 - (C_2 + C_3 x_2) x_2$$

The quadratic is evaluated for a given value of x if a code is positive.

USAGE: DIMENSION X(1), Y(1), C(3)

CALL QRAT(X, Y, IXY, KODE, XI, YI, C, NU)

Input:

X = Location of the x-coordinate of the first point.

Y = Location of the y-coordinate of the first point.

IXY = Skip number for X and Y. IXY = location of $w_{i+1}$ minus the location of $w_i$, where w is either x or y and i = 1 or 2. For example, IXY = 1 if X and Y are each dimensioned 3. Or if an array $W = (x_1, y_1, x_2, y_2, x_3, y_3)$ then CALL QRAT (W, W(2), 2, ...) may be used to avoid shifting $x_i$ and $y_i$ to separate arrays

KODE = If KODE > 0, the quadratic (if found) will be evaluated at x = XI and the result stored in YI. Otherwise, XI and YI are not used.

XI = See KODE.

| | | |
|---|---|---|
| Output: | YI | = See KODE. |
| | C | = Array of quadratic coefficients $C = (C_1, C_2, C_3)$. |
| | NU | = An error code that is zero if QRAT is successful. |

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = 1 if any two of the given points have the same x-coordinate.

RESTRICTIONS: IXY may not be zero; it may be negative if the point coordinates are stored in descending core locations.

STORAGE: $157_{10} = 235_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine READ |
| PURPOSE: | To provide, as options, three methods of inputting an array of values. |

METHOD:

The subroutine allows control over the form of input to the program by means of a literal word on a data card. Two words are stored in the program Data Dictionary. The first control word is CONSTANT, of which the first six letters CONSTA are stored. The second word is GIVEN. If the word on the data card matches either of the two control words, transfer in the program is made accordingly. The third option is provided by arbitrary letters in the first six columns of the input card. The three options provided are:

1) If the first data card contains CONSTANT, the program expects one additional data card on which the first field (columns 1-10) contains a value. This value is then set in the array S.

2) If the first data card contains GIVEN, the program expects to read an array of values off a specified logical tape unit.

3) If the data card contains any arbitrary alphanumeric characters, the program expects to read an array of values from data cards punched in a seven-field, ten-digit format.

USAGE:

CALL READ (NTAPEI, NTAPEO, NTAPEC, NM, S)

DIMENSION S(NM), DICT(2)

| | | |
|---|---|---|
| Input: | NTAPEI | = Logical number of input tape. |
| | NTAPEO | = Logical number of output tape. |
| | NTAPEC | = Logical number of scratch tape, if option 2. |
| | NM | = Number of values in the input array. |
| Output: | S | = Array of values. |

SUBPROGRAM CALLED: INTURP

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $131_{10} = 203_8$

SUBJECT: FORTRAN IV Subroutine REDUCE

PURPOSE: To form the "reduced" aerodynamic influence coefficients matrix and three additional matrices in preparation for use in later links.

METHOD: The aerodynamic influence coefficients matrix is composed of the normal velocity components induced by a unit pressure difference on each panel.

The matrix may be written in terms of the sub-matrices

$$\begin{bmatrix} [A_{BB}] & [A_{BW}] \\ [A_{WB}] & [A_{WW}] \end{bmatrix} \cdot \begin{Bmatrix} p_B \\ p_W \end{Bmatrix} = \begin{Bmatrix} -\ n_{BWS} \\ \frac{dZc}{dx} - (\alpha) - n_{WB} \end{Bmatrix}$$

where

$A_{BB}$ = The influence on the body due to the body

$A_{WB}$ = The influence on the wing due to the body

$A_{BW}$ = The influence on the body due to the wing

$A_{WW}$ = The influence on the wing due to the wing

$n_{BWS}$ = The normal velocity component on the body due to wing sources

$\frac{dZc}{dx}$ = The slope of the wing panels

$\alpha$ = The angle of attack

$n_{WB}$ = The normal velocity component on the wing due to the body

The matrix equation is solved for $p_B$ and $p_W$.

$$\{p_B\} = -\ [A_{BB}]^{-1}\left\{\{n_{BWS}\} + [A_{BW}]\{p_W\}\right\}$$

$$\{p_W\} = [A_R]^{-1}\left\{[A_{WB}][A_{BB}]^{-1}\{n_{BWS}\} + \left\{\frac{dZc}{dx} - \alpha - n_{BWS}\right\}\right\}$$

where $[A_R] = \left[[A_{WW}] - [A_{WB}][A_{BB}]^{-1}[A_{BW}]\right]$

is referred to as the "reduced" aerodynamic matrix.

The "reduced" matrix is then written on logical tape NTAPEE.

The additional matrices formed are

$$[D] = [A_{WB}][A_{BB}]^{-1}$$

and

$$[E] = [A_{BB}][A_{BW}]$$

Three matrices are written on logical tape NTAPEF in the order $[A_{BB}]^{-1}$, $[D]$, $[E]$ with an end-of-file written after each matrix.

USAGE: CALL REDUCE

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION: AB(100), ABB(100, 100), AWB(110, 100), ABW(100, 25), AWW(100, 25), E(100, 25), D(110)

EQUIVALENCE (ABB, AWB)

Input:

NWING = Number of wing panels.

NBODY = Number of body panels.

NTAPEA = Logical number of tape from which the aerodynamic influence coefficients are read.

NTAPED = Logical number of tape from which $[A_{WB}]$ and $[A_{BB}]^{-1}$ are read.

NTAPEE = Logical number of tape on which the reduced aerodynamic influence coefficients matrix is written.

NTAPEF = Logical number of tape on which $[A_{BB}]^{-1}$, $[D]$, and $[E]$ are written.

AWB = Matrix $[A_{WB}]$

ABB = Matrix $[A_{BB}]^{-1}$

ABW = Matrix $[A_{BW}]$

AWW = Matrix $[A_{WW}]$

Output:

AWW = Matrix $[A_R]$

ABB = Matrix $[A_{BB}]^{-1}$

D = Matrix $[D]$

E = Matrix $[E]$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: NWING $\leq$ 110

NBODY $\leq$ 100

STORAGE: $19564_{10} = 46152_8$

SUBJECT: FORTRAN IV Subroutine RICH3A

PURPOSE: Given a sparse array of points on a space curve, to compute a dense array.

METHOD: A given tolerance, CHD, regulates the number of additional points to be generated on the space curve. Between adjacent points in the dense array, the maximum distance between arc and chord will be approximately less than CHD. (However, if CHD $\leq 0$ or if there are fewer than four points in the sparse array, no interpolation takes place and the dense array will be identical to the sparse array.)

The given sparse-array points are projected into a plane. Additional points are interpolated in this plane by subroutine ENRYCH (using the tolerance CHD). An inverse projection is then performed, mapping the two-dimensional points back to the original three-space.

A more detailed description of the enriching process follows. As a first step the sparse array of points is transformed to an X', Y', Z' coordinate system. The transformation is determined by a given code, KD. Let $P_1$ $(x_1, y_1, z_1)$ be the first point in the given sparse array P, and $P_n$ $(x_n, y_n, z_n)$ be the last point. If KD = 0, the X' Y' Z' origin is at $P_1$, and the X' axis passes through $P_n$; the X' Z' plane contains the point in P farthest from the line $P_1$ $P_n$. If KD = 1 or 2, the X' Y' Z' origin is at $(x_1, y_1, 0)$, and the X' axis passes through the point $(x_n, y_n, 0)$; the Z' axis is parallel to the Z axis. In the next step, the X' Z' coordinates of the points are treated as two-dimensional points by subroutine ENRYCH to produce a dense array of points. To find the corresponding Y' coordinate of each point, a two-dimensional interpolation is made by subroutine BITURP: the interpolation table contains the sparse array X' and Y' coordinates, with X' as the independent variable. For each dense array point the Y' coordinate is interpolated from the table. The interpolation is linear if KD = 1, or biquadratic if KD = 0 or 2. In the final step, the dense array X', Y', Z' coordinates are transformed to the original X, Y, Z system.

USAGE: DIMENSION P(3, N), Q(3, N), A(3, MAX/3), NU(3)

CALL RICH3A (N, P, Q, KD, MAX, CHD, NA, A, NU)

Input: N = Number of points in P.

P = Array of given points (sparse array).

| | | |
|---|---|---|
| | KD | = Code (see METHOD, above). |
| | MAX | = Maximum number of elements that may be stored in the A-array. |
| | CHD | = Chord-height tolerance that determines the spacing of interpolated points (see ENRYCH |
| | NU | = Error indicator array: NU(1) is not used on input. NU(2) and NU(3) are used only by ENRYCH and BITURP. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written. |
| Output: | Q | = Scratch array that may occupy the same storage space as P if P need not be preserved. |
| | NA | = Number of points in the A-array. |
| | A | = Array of enriched points (dense array). |
| | NU | = Error indicator array: NU(1) = 0 if success. NU(3) is discussed above. |

SUBPROGRAMS CALLED:

UVECN
VCROS
DMAXL
ENRYCH
BITURP
TROTPT
SQRT (Built-in function)
TRAV

ERROR RETURNS:

NU = -2, if KD = 0 and the endpoints of P are coincident. NU = -1 (corresponding to error code 3 from ENRYCH) if more than 200 interpolated points would be required between two points in P: either CHD is too small or the points in P (after projection) are ill-conditioned. RICH3A adds no interpolated points in this interval, and continues. NU = 1, 4, 5, or 6 if MAX is too small. NU = 2 (error 2 from ENRYCH) if four consecutive points in P (after

transformation to the X', Y' system) are not strictly monotonic in either X' or Y'. NU = 7 if KD = 1 or 2 and the endpoints of P have the same X and Y coordinates. NU = 11, 12, 13, or 14 (BITURP errors 1, 2, 3, or 4) if an error was detected during the inverse projection (see METHOD, above); this can happen only if the given sparse array is ill-conditioned. NU = 21 if an error occurs when finding the projection plane.

RESTRICTIONS: After transformation to the X', Y', Z' system (see METHOD, above), the given sparse array points must be monotonic in X'; no two X' coordinates may be the same. Since only the projections of the given points on the X'-Z' plane are enriched, too few points may be generated if the given points do not lie approximately in the plane.

STORAGE: $509_{10} = 775_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine RICHNA |
| PURPOSE: | To enrich (interpolate additional points) a set of body meridian lines or wing percent lines. |

METHOD:

A given array A of length MAX contains M arrays of N three-dimensional points each. Each array will be called a percent line. Since array A is also used for output, all the points are first shifted to the end of A. A given code, KOD, determines the type of interpolation (see subroutine RICH3A). If KOD $\neq$ 0, all points in each percent line are checked: a comment is written for each point having X and Y coordinates that are nearly the same (within a given tolerance EPS) as those of the previous point but for which the Z-coordinates are different. Then all percent lines are screened by function NWEED to remove any points too close to the preceding point; If KOD = 0, all three coordinates must nearly match the previous point before a point is removed. But if KOD > 0, only the X and Y coordinates must nearly match; thus any points for which a comment was written will be removed.

(NOTE: KOD = 0 for body meridian lines; KOD = 0, 1, or 2 for wing percent lines.) If a given tolerance CHD is less than or equal to 0, no additional points will be interpolated. A "header" or "table of contents" is inserted at the beginning of array A; this is followed by the point coordinates. The format of the completed array A is given in Appendix D.

USAGE:

DIMENSION A(MAX), P(M), NU(3)

CALL RICHNA (A, MAX, M, N, P, EPS, CHD, KOD,
NA, NU)

Input:

| | | |
|---|---|---|
| A | = | Array of M percent lines of N points each. |
| MAX | = | Length of array A. |
| M | = | Number of percent lines. |
| N | = | Number of points in each percent line. |
| P | = | Array of labels that are angles (if body meridian lines) or percent values (if wing percent lines); these are used in the "header" of the final array (see Appendix D). |
| EPS | = | Tolerance of test for coincident points. |

CHD = Chord-arc tolerance that controls the number of interpolated points. See subroutine RICH3A.

KOD = Code which determines type of interpolation (see KD in subroutine RICH3A).

NU = Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected in some lower-level subroutine; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) =(NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: A = Array of M enriched percent lines, with "header" that contains a label, number of points, and starting location in A for each percent line. See Appendix D for format.

NA = Total number of <u>points</u> in array A. The total number of <u>elements</u> used in A is 3 * (M + NA).

NU = Error indicator array: NU(1) = 0 if no errors are detected. NU(3) may have been reduced by a lower-level subroutine if an error occurred.

SUBPROGRAMS CALLED: NWEED
RICH3A

ERROR RETURNS: NU = -1 if NWEED detected an error; this could be caused by a very large value of EPS. For any other nonzero value of NU, see RICH3A.

RESTRICTIONS: MAX $\geq$ 3 * M * (N + 1)

STORAGE: $400_{10} = 620_8$

SUBJECT: FORTRAN IV Subroutine RITE

PURPOSE: To output the x, y, and z components of velocity induced by the various singularities.

METHOD: If a printout of the velocity components is requested, the x, y, and z components of velocity are written on the output tape. The components are identified in the printout as the x component, AXIAL(U); the y component, TRANSVERSE(V); and the z component, VERTICAL(W). By an input code, the proper identification relating the velocity components to the singularity is also printed.

USAGE: Call RITE (NFMT, NTAPEO, NP, NROW, NCOL, THETA, U, V, W)

DIMENSION NFMT(7), U(NP), W(NP)

Input: NFMT control code

| | | |
|---|---|---|
| NFMT(I) (I = 1, 7) | = | 0, output of velocity components not requested. |
| NFMT(1) | = | 1, velocity components on body panels resulting from body panel pressure singularities. |
| NFMT(2) | = | 2, velocity components on body panels resulting from wing panel pressure singularities. |
| NFMT(3) | = | 3, velocity components on body panels resulting from wing sources. |
| NFMT(4) | = | 4, velocity components on wing panels resulting from wing panel pressure singularities. |
| NFMT(5) | = | 5, velocity components on wing panels resulting from wing sources. |
| NFMT(6) | = | 6, velocity components on wing panels resulting from body panel pressure singularities. |
| NFMT(7) | = | 7, velocity components on wing panels body line sources and doublets. |
| NTAPEO | = | Logical number of output tape. |
| NP | = | Number of wing or body panels. |

| | | |
|---|---|---|
| NROW | = | Number of rows of panels on wing or body. |
| NCOL | = | Number of chordwise columns of panels on wing or number of longitudinal columns of panels on body. |
| THETA | = | If body case, the $\theta$ angles around the body of the data output points. |
| U | = | x component of velocity. |
| V | = | y component of velocity. |
| W | = | z component of velocity. |

SUBPROGRAMS CALLED: OUTB
OUTW

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $409_{10} = 631_8$

SUBJECT: FORTRAN IV Subroutine RRAT

(See Subroutine QRAT)

SUBJECT: FORTRAN IV Subroutine SAVTAP

PURPOSE: To write the aerodynamic matrices for a given configuration and Mach number on a logical tape that can be saved and used for later computer runs.

METHOD: The aerodynamic matrices are read from several logical tapes as indicated below and written as one file on logical tape NTAPEC.

For wing-alone configurations, the matrices are written in the following order:

| Previous tape | Matrix | Description |
|---|---|---|
| 1. NTAPEA | $\left[A_{WW}\right]$ | Aerodynamic influence coefficients due to wing sources. |
| 2. NTAPEA | $\left[WW\right]$ | Drag minimization. |
| 3. NTAPEA | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift. |
| 4. NTAPEA | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift and pitching moment. |
| 5. NTAPEB | $\left[U_{WW}\right]$<br>$\left[V_{WW}\right]$<br>$\left[W_{WW}\right]$ | Velocity components due to wing sources. |
| 6. NTAPEB | $\left[U_{WW}\right]$<br>$\left[V_{WW}\right]$<br>$\left[W_{WW}\right]$ | Velocity components due to wing surface vortices. |
| 7. NTAPEE | $\left[A_{WW}\right]$ | Aerodynamic influence coefficients due to wing surface vortices. |
| 8. NTAPEE | $\left[A_{WW}\right]^{-1}$ | Inverse of matrix of aerodynamic influence coefficients due to wing surface vortices. |

For wing-body configurations, the matrices are written in the order:

| Previous tape | Matrix | Description |
|---|---|---|
| 1.-5. (same as wing-alone configurations) | | |
| 6. NTAPED | $[U_{BB}]$ $[V_{BB}]$ $[W_{BB}]$ | Velocity components on the body due to body surface vortices. |
| 7. NTAPED | $[U_{WB}]$ $[V_{WB}]$ $[W_{WB}]$ | Velocity components on the wing due to body surface vortices. |
| 8. NTAPED | $[U_{BW}]$ $[V_{BW}]$ $[W_{BW}]$ | Velocity components on the body due to wing surface vortices. |
| 9. NTAPED | $[U_{WW}]$ $[V_{WW}]$ $[W_{WW}]$ | Velocity components on the wing due to wing surface vortices. |
| 10. NTAPEE | $[A_R]$ | "Reduced" aerodynamic matrix. |
| 11. NTAPEE | $[A_R]^{-1}$ | Inverse of "reduced" aerodynamic matrix. |
| 12. NTAPEF | $[A_{BB}]^{-1}$ | Inverse of matrix of aerodynamic influence coefficients. |
| 13. NTAPEF | $[D]$ | Product matrix. $[D] = [A_{WB}][A_{BB}]^{-1}$ |

| Previous tape | Matrix | Description |
|---|---|---|
| 14. NTAPEF | [E] | Product matrix. $\left[E\right] = \left[A_{BB}\right] \cdot \left[A_{BW}\right]$ |

This subroutine is not used for body-alone configurations.

USAGE: CALL SAVTAP (ISAVET)

Input: ISAVET = Code that indicates if matrices are to be saved. If ISAVET $>$ 0 matrices are written as one file on logical tape NTAPEC as indicated above. If ISAVET $\leq$ 0, a dummy file is written on logical tape NTAPEC.

$\left.\begin{array}{l}\text{NTAPEA}\\ \text{NTAPEB}\\ \text{NTAPED}\\ \text{NTAPEE}\\ \text{NTAPEF}\end{array}\right\}$ = Logical tape numbers from which aerodynamic matrices are read.

NTAPEC = Logical tape number on which aerodynamic matrices are written and saved.

SUBPROGRAMS CALLED: TTAPE

FSF

ERROR RETURNS: None

RESTRICTIONS: This subroutine cannot be used for body-alone configurations.

STORAGE: $1212_{10} = 2274_8$

SUBJECT: FORTRAN IV Subroutine SCAMP4

PURPOSE: Given a set of n points $(x_i, y_i)$ whose abscissae form a strictly monotone sequence, a first or second derivative at $x_1$, and a first or second derivative at $x_n$, to find the smoothest possible curve passing rigorously through the given points, satisfying the specified boundary conditions, and possessing continuous first and second derivatives. The criterion for smoothness is the minimization of the integral of the square of the second derivative, from $x_1$ to $x_n$, over all functions having the stated properties. Accordingly, the curve found is a chain of cubics, i.e., a separate cubic defined on each interval $(x_i, x_{i+1})$. The coefficients of each such cubic are explicitly found in the form:

$$y = C_0 + C_1x + C_2x^2 + C_3x^3$$

METHOD: The most economical (in time and space) and most accurate method of finding such a chain of cubics is to solve first for the n slopes $y_i$ of the curve. This is done by the composite cubic subroutine COMCU, which solves an nth order linear system, the coefficient matrix of which is tridiagonal. Having found the slopes at each of the n given $x_i$, one can determine the coefficients of each cubic separately by using CUBIC2, which finds the cubic through two points, being given the slope at each. The coefficients of all the n - 1 cubics can be obtained by using the subject routine (SCAMP4) which serves as a vehicle for calling COMCU (once) and CUBIC2 (n-1 times). SCAMP4 has an option to compute the required boundary conditions (first or second derivatives at the end points) if these are not known by the calling program; in this case, the computation of <u>first</u> derivatives at $x_1$ and $x_n$ is recommended.

The cubic coefficients found by SCAMP4 are either stored in a 4 by n - 1 array or are arranged in the composite curve format, i.e. in a single linear array where each segment is specified by a block of seven consecutive words: $x_i$, $x_{i+1}$, 3., $c_0$, $c_1$, $c_2$, $c_3$. The calling program should dimension the coefficient array as a doubly subscripted variable in the former case and singly subscripted in the latter case.

USAGE: CALL SCAMP4 (X, Y, N, NDA, NDB, DA, DB, C, S, M)

DIMENSION X(N), Y(N), C(4, N-1) or C(7, K)

where K = N - 1

| | | | |
|---|---|---|---|
| Input: | X | = | Array of x-abscissae. |
| | Y | = | Array of y-ordinates. |
| | N | = | Number of points. |
| | NDA | = | The order (1 or 2) of the derivative to be given at X(1). If derivative is to be computed by SCAMP4, NDA $\leq$ 0. |
| | NDB | = | The order of the derivative to be given at X(N). Similar to NDA. |
| | DA | = | The value of the derivative at X(1). If derivative is to be computed by SCAMP4, leave blank. |
| | DB | = | The value of the derivative at X(N). Similar to NDA. |
| | M | = | Code. |
| | | $\neq$ | 12, if the cubic chain coefficients are to be stored in a doubly dimensioned 4 x (N - 1) array. |
| | | = | 12, if the cubic chain coefficients are to be stored in a singly dimensioned array. |
| Output: | C | = | Array of cubic chain coefficients. |
| | S | = | Array of first derivatives. |
| | M | = | Error return. |
| | | = | 0 — success |
| | | $\neq$ | 0 — error detected |

SUBPROGRAMS CALLED:

COMCU
CUBIC2
DERIV1
DERIV2

ERROR RETURNS: $M = -1$, indicates $N < 2$. $1 \leq M \leq 7$, indicates an error return from COMCU. M large indicates error return k on the $j^{th}$ call to CUBIC2 ($M = 100 \cdot j + k$).

STORAGE: $288_{10} = 440_8$

SUBJECT: FORTRAN IV Subroutine SIMUN3

PURPOSE: Given a set of n points $(x_i, y_i)$, to numerically calculate using Simpson's (parabolic) rule the value of the integral:

$$\int_{x_1}^{x_n} y \, dx.$$

METHOD: Simpson's rule for a single interval containing three equally spaced points can be written:

$$\int_{x_1}^{x_3} y \, dx \sim \frac{\Delta x}{3} (y_1 + 4y_2 + y_3)$$

If, in fact, there are m successive subintervals each containing three points for a total of 2m + 1 equally spaced points, this formula can be rewritten:

$$\int_{x_1}^{x_{2m+1}} y \, dx \sim \frac{\Delta x}{3} (y_1 + 4y_2 + 2y_3 \ldots + 2y_{2m-1} + 4y_{2m} + y_{2m+1})$$

The set of input points $(x_i, y_i)$ may or may not consist of some odd number of equally spaced points as required for Simpson's rule. Subroutine SCAMP4 is first used to compute the coefficients of a chain of cubics fit to these input points. A second set of 2m + 1 equally spaced points $(\bar{x}_i, \bar{y}_i)$ is then obtained by using the cubic chain coefficients provided by SCAMP4 and interpolating on the set of input points. If for some $\bar{x}_i$ :

$$x_j < \bar{x}_i \leq x_{j+1}$$

then,

$$\bar{y}_i = c_{1,j} + c_{2,j} \bar{x}_i + c_{3,j} \bar{x}_i^2 + c_{4,j} \bar{x}_i^3$$

where $c_{1,j}$, $c_{2,j}$, $c_{3,j}$ and $c_{4,j}$ are the coefficients of the $j^{th}$ cubic polynominal. A numerical value for the intergral in question is then calculated by directly applying Simpson's rule:

$$\int_{x_1}^{x_{2m+1}} y \, dx \sim \frac{\Delta x}{3} \left( \bar{y}_1 + 4 \sum_{i \text{ odd}} \bar{y}_i + 2 \sum_{i \text{ even}} \bar{y}_i + \bar{y}_{2m+1} \right)$$

USAGE: CALL SIMUN3 (X, Y, M, A, K)

DIMENSION X (M), Y (M)

| | | | |
|---|---|---|---|
| Input: | X | = | Array of x-abscissae of input points. |
| | Y | = | Array of y-ordinates of input points. |
| | M | = | Number of input points. |
| Output: | A | = | Value of integral. |
| | K | = | Error return |
| | | = | 0 — success |
| | | ≠ | 0 — error detected. |

SUBPROGRAMS CALLED: SCAMP4

MOD, FLOAT (Built-in functions)

ERROR RETURNS: Three types of errors are detected and indicated. If fewer than three points are input, K = 1. If SCAMP4 detects an error in fitting a chain of cubics to the set of input points, K = 2. If an error occurs in calculating the integral, K = 3. Otherwise, K = 0.

RESTRICTIONS: $3 \leq M \leq 55$

STORAGE: $628_{10} = 1164_8$

SUBJECT: FORTRAN IV Subroutine SIZE

PURPOSE: To determine the number and sizes of the partitions of a matrix, given the size of the matrix and a maximum partition size.

METHOD: The subroutine uses the following sequence of FORTRAN instructions

```
         NPMIN    = 2
         NT       = MM
         NPART    = MAX0 (NPMIN,  (NT - 1)/NMAX + 1)
         N        = NPART
10       NSIZE(N) = NT/N
         NT       = NT - NSIZE(N)
         N        = N - 1
         IF(N)  20, 20, 10
20       RETURN
```

USAGE: CALL SIZE (MM, NMAX, NPART, NSIZE)

DIMENSION NSIZE (NPART)

| | | | |
|---|---|---|---|
| Input: | MM | = | Size of matrix. |
| | NMAX | = | Maximum partition size. |
| Output: | NPART | = | Number of partitions. |
| | NSIZE | = | Array of partition sizes. |

SUBPROGRAM CALLED: MAX0 (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: None

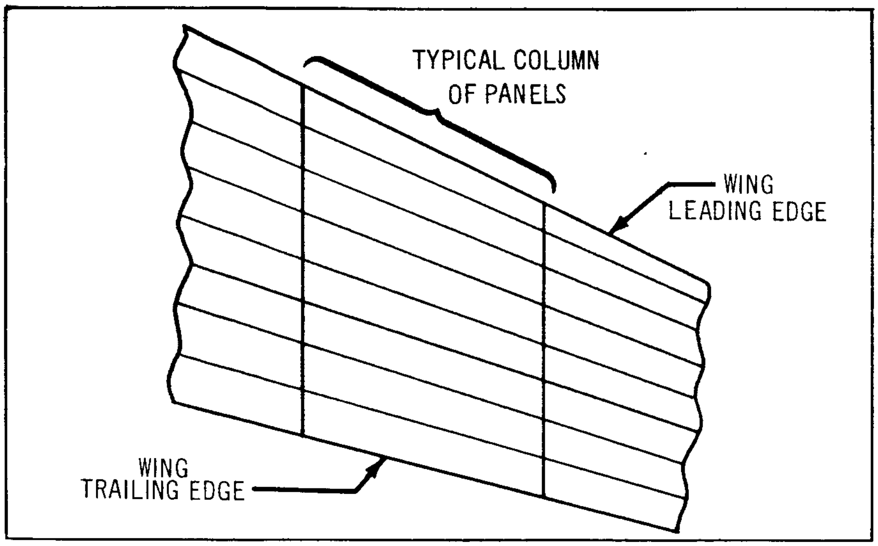
STORAGE: $74_{10} = 112_8$

SUBJECT: FORTRAN IV Subroutine SLOPEW

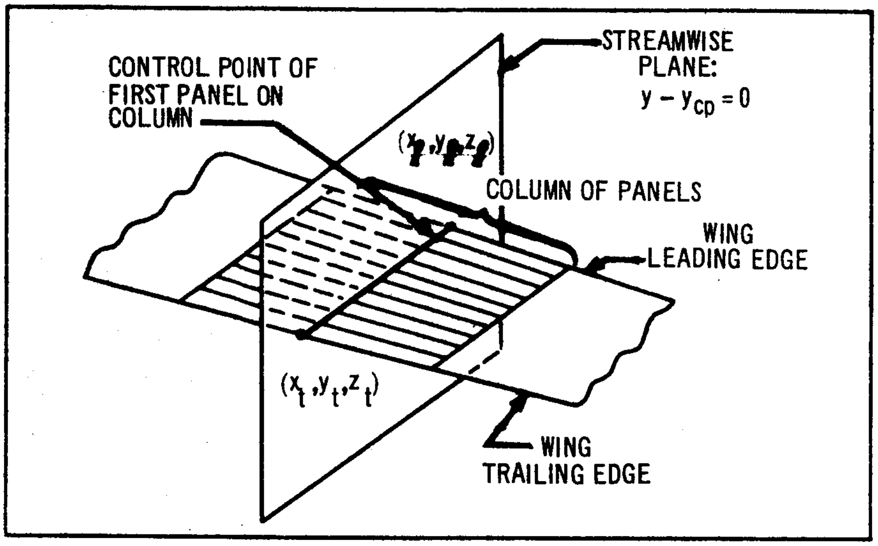
PURPOSE: To calculate camber and thickness ordinates and camber and thickness slopes for an airfoil.

METHOD: Program input can consist of single airfoil (or multiple airfoils) for which both an upper and lower surface are defined (see subroutine INPUTW). If a single airfoil is given, this airfoil is scaled accordingly and used to calculate camber and thickness for each column of panels on the wing. If multiple airfoils are given, each airfoil is reduced to a unit airfoil, scaled accordingly and used for camber and thickness calculations on but a single column of wing panels. Note that, for the latter case, as many airfoils as panel columns must be given on input cards.

For a given unit airfoil and the corresponding panel column, the program locates a local streamwise chord on the column, computes the chord length, and uses this value to scale the airfoil. For example, a typical panel column is represented below:




The program constructs a streamwise plane through the control point of the first panel on the column and intersects this plane with the wing leading and trailing edges. The panel side edges are streamwise; thus this construction is always possible and, because the control points for a given column of panels lie on the same streamwise chord, only one such construction is necessary. This is shown in the following sketch:




A chord length to be associated with this panel column is then calculated,

$$c = x_t - x_\ell$$

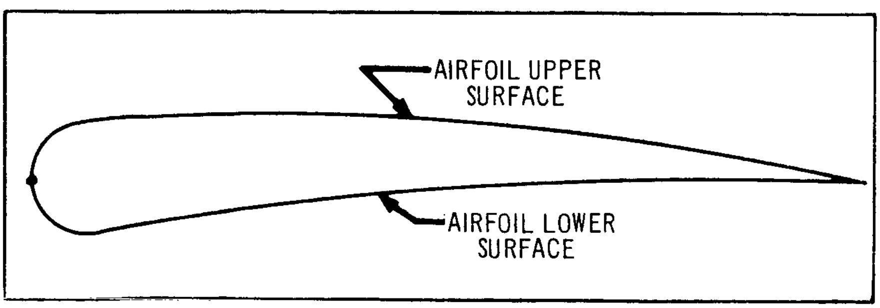
and this chord length is used to scale the airfoil so that,

$$x_k = x_{k_o} \cdot c$$

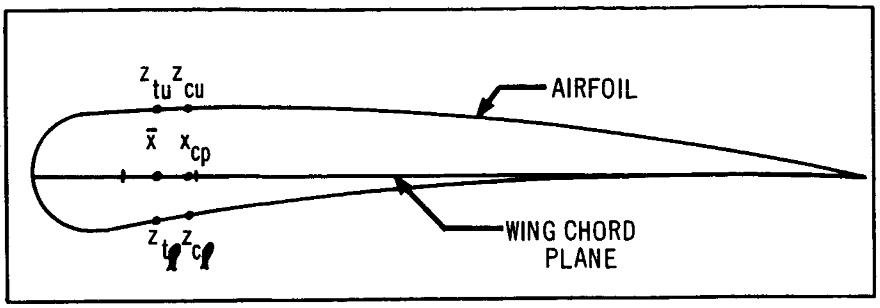
$$z_k = z_{k_o} \cdot c + \zeta$$

for which $x_{k_o}$, $z_{k_o}$ are the x- and z-coordinates of the unscaled, unit airfoil, $x_k$, $z_k$ are the coordinates of the scaled airfoil and $\zeta$ is the constant z-value for the wing.

Having scaled the airfoil to "fit" the wing for a particular column, the program then calculates camber and thickness ordinates for each panel of that column. To satisfy a condition of the program, the camber ordinate is computed at the panel control point and the thickness ordinate at the panel centroid. Because both the upper and lower surfaces of an airfoil are given as input (see sketch),



the program must interpolate on each surface for both the camber and thickness ordinate. As a result, four values are calculated




and used in camber and thickness calculations. For the camber ordinate,

$$z_{cam} = \frac{1}{2}(z_{cu} + z_{c\ell})$$

and for thickness,

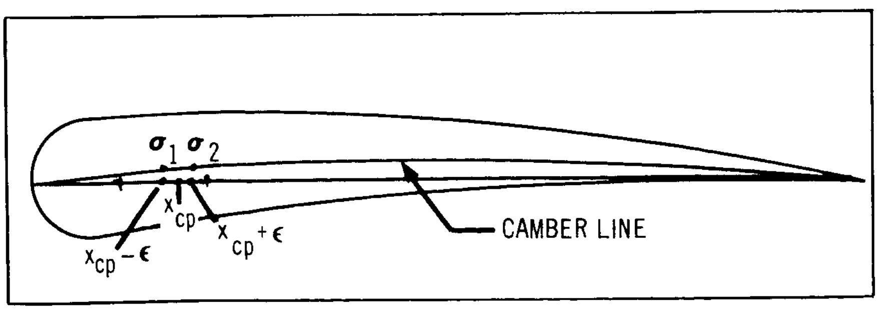
$$z_{thk} = \frac{1}{2}(z_{tu} - z_{t\ell}).$$

Using the camber and thickness ordinates that have been calculated, the program computes the camber and thickness slopes. As the camber is evaluated at the panel control point and the thickness at the centroid, the program constructs two distinct tables:

1) $z_{cam}$ versus $x_{cp}$
2) $z_{thk}$ versus $\bar{x}$

that are used for these calculations.

To obtain the camber slope at the $x_{cp}$ for a given panel, the program evaluates the camber curve (1) at $x_{cp} - \epsilon$ and $x_{cp} + \epsilon$



for which

$$\epsilon = \frac{c}{10^5}$$

and c is the chord length.

The values $\sigma_1$, $\sigma_2$ of the curve at these two points are used to compute the camber slope,

$$\alpha_{cam} = \tan^{-1}\left(\frac{\sigma_2 - \sigma_1}{2\epsilon}\right)$$

Similarly, to obtain the thickness slope at the $\bar{x}$ of a given panel, the thickness curve (2) is evaluated at $\bar{x} - \epsilon$ and $\bar{x} + \epsilon$, and the values $\tau_1$ and $\tau_2$ of the curve at these two points are used,

$$\alpha_{thk} = \tan^{-1}\left(\frac{\tau_2 - \tau_1}{2\epsilon}\right)$$

As a last step the program smooths the camber and thickness slopes. The calculations of the slopes for a given column of panels result in two tables:

1) $\alpha_{cam}$ versus $x_{cp}$
2) $\alpha_{thk}$ versus $\bar{x}$

and each of these tables is smoothed through use of the subroutine MEAN.

The above calculations are used to obtain camber and thickness ordinates and camber and thickness slopes for panels of a column of the wing in all but two cases,

1) The inboard column on the wing for a body-wing case having a body contour in the intersection region;
2) The outboard wing column for wing having a nonstreamwise outboard wing edge.

For these two cases, the calculations are changed slightly because the edges of an arbitrary panel are not streamwise and the control points of all panels of the column do not lie on the same streamwise chord. A local streamwise chord must be constructed and a chord length determined for each panel in the column. The unit airfoil is scaled to "fit" the wing at the $y_{cp}$ of each panel; then the camber and thickness are calculated.

USAGE:

COMMON (See subroutine OPCAMI for unlabeled COMMON description)

COMMON /COM1/ (See subroutine PANEL)

Input: KOPTF
NUMS
KSTART
KEND
NPER
NPER1
NPLANE
NPLN1

COMMON /COM2/ (See subroutine WING)

Input: NPTS
X
Y
Z
SLOPE
XCEN
XCON
YCON
XFOIL
ZFOIL
XNUM

Output: ZCEN
ZCON
THETA
ALPHA

CALL SLOPEW

SUBPROGRAMS CALLED:

ATAN (Built-in function)
BITURP
MEAN
POLXN
CHECK

ERROR RETURNS: Error messages indicate whether error occurred in use of subroutine BITURP to interpolate on airfoil surfaces or on camber and thickness curves. An error message also indicates if the airfoil x-ordinates are not defined as a (strictly) monotonic increasing sequence.

STORAGE: $2272_{10} = 4340_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine STRM1 |
| PURPOSE: | To integrate a three-dimensional streamline around a supersonic wing-body combination. |
| METHOD: | LACKEY obtains velocities which are interpreted as derivatives of the streamline. The DEDIS package is used to perform an Adams-Moulton predictor-corrector variable stepsize integration. (See DEDIS) |
| USAGE: | COMMON /INTGA/ DUM (3500) |

CALL STRM1 (X, Y, Z, XMAX, XMIN, XDELT, DXMAX, DXMIN, IND1, IND2)

| | | |
|---|---|---|
| Input: | x | = x-coordinate of starting point of streamline. |
| | y | = y-coordinate of starting point of streamline. |
| | z | = z-coordinate of starting point of streamline. |
| | XMAX | = Maximum value which the x-coordinate of the streamline is allowed to attain. |
| | SMIN | = Minimum value which the x-coordinates of the streamlines is allowed to attain. |
| | XDELT | = Initial stepsize. |
| | DXMAX | = Maximum stepsize allowed. |
| | DXMIN | = Minimum stepsize allowed. |
| | IND1 | = 1. Negative running streamline. |
| | | = 2. Positive running streamline. |
| | IND2 | = 1. Interpolated output (see output). |
| | | = 2. Actual points determined by integration. |

Output: On user option, the streamlines are written on the output tape in step increments of XDELT or the actual points are determined by the variable stepsize integration are written.

| | |
|---|---|
| SUBPROGRAMS CALLED: | DED<br>DES<br>DEI<br>BJTURP<br>LACKEY<br>SQRT (Built-in function ) |
| ERROR RETURNS: | None |
| RESTRICTIONS: | If the number of points determined by the integration routine exceeds 200 before reaching one of the x-coordinate extrema (DXMAX, or DXMIN), the integration is terminated and results are output at that point. |
| STORAGE: | $734_{10} = 1336_8$ |

SUBJECT: FORTRAN IV Subroutine TAN

(See subroutine FTAN)

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TDUMP |
| PURPOSE: | To read the binary tape produced by the Geometry Definition section and to write its contents on an output tape. |
| METHOD: | A given logical array, which contains tape record status information, is checked. If there are no valid records on the binary tape, control is returned to the calling program. Otherwise, each of the 18 records is processed. The tape format is given in Appendix D. |
| USAGE: | LOGICAL LGDEF(3, 6)<br>CALL TDUMP (LO, LTAPE, LGDEF)<br>Input: LO = Output tape number.<br>LTAPE = Binary tape number.<br>LGDEF = See subroutine GEOMD. |
| SUBPROGRAM CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | Records must not contain more than 2000 words. |
| STORAGE: | $2175_{10} = 4177_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TFLAT |
| PURPOSE: | To transform body and wing to a new coordinate system, to flatten the wing, and to find the intersection of each wing percent line with the body surface. |

METHOD:

Three types of cases may be handled: body alone, wing alone, or wing-body combination. If a body surface is involved, it is assumed to be symmetrical about the X-Z plane; only half of it is given. Then a new x, y, z coordinate system is set up with origin at the centroid of forward end of the body and the x-axis passing through the aft-end centroid; the z-axis lies in the X-Z plane. In the new coordinate system, the body ends (unless of zero radius) are not necessarily parallel to the y-z plane; this is corrected by a linear adjustment of all x-coordinates of the body points. Also, the new body length (distance between body end centroids) may be different than the original length. The original set of body defining stations is converted to the new coordinate system by a transformation of the form $x = \alpha X + \beta$. The body radius and centroid are computed at each of these stations. In a body alone case, the radius and centroid are then found by interpolation at a given number of equally spaced stations; otherwise, this calculation is deferred.

In a wing-body combination, the wing is also transformed to the x, y, z coordinate system. Then the intersections of the wing leading and trailing edges with the body surface are computed; let $z_A$ be the average of the z-coordinates of these two points. The wing is effectively flattened by changing each z-coordinate of the wing to $z_A$. The body is then intersected by each wing percent chord line of the flattened wing.

The body radius and centroid calculation can now be resumed. A set of equally spaced stations is chosen between the forward end of the body and the station at which the wing leading edge intersects the body; a similar set is selected from there on aft. The calculation of the interval between stations in each region is subject to the constraints that the total number of stations must agree with a given number and that the two intervals must be approximately the same in each region.

In a wing alone case, the transformation is omitted. However, the wing is flattened: $z_A$ is taken

as the average of the z-coordinates of the first point in the leading- and trailing-edge percent lines.

The body and wing definitions are on a binary tape, LA, which has a format as described in Appendix D. The body is given by meridian lines, the wing by percent chord lines; meridian and percent lines are described by sets of three-dimensional points (see subroutine WBXUL). One data card is required; it is read with FORMAT (5F10.0) as follows:

| Columns | Symbol | Explanation |
|---|---|---|
| 1-10 | XNRX | Number of body stations at which body radius and centroid are desired. If this number is less than 3 or greater than 50, it is set equal to 50. |
| 11-20 | CHD | A tolerance passed on to subroutine TFLAT1, and then to BODCR, for eventual use by subroutines ENRYCH and OPTIM3. CHD is used only to find body centroids, as follows: A section, normal to the x-axis, is taken through the body. The section has one point for each meridian line. If CHD $\leq 0$, the centroid is found assuming the section to be polygonal. If CHD $> 0$ and there are at least four meridian lines, additional section points are generated by ENRYCH on a smooth curve through the vertices of the polygon: a sufficient number of points will be generated so that the difference between the chord of the resultant polygon and the arc of the smooth curve is (approximately) less than CHD. A positive value of CHD should not be used unless there are at least eight meridian lines on a non-symmetrical (about a horizontal plane) body. |

| Columns | Symbol | Explanation |
|---|---|---|
| 21-30 | BCODE (1) | An interpolation code used by BITURP to find body radius and $z_c$, (z-coordinate of the body centroid) at values of x between defining stations. BCODE (1) = 1 for linear interpolations, = 2 for biquadratic interpolation. |
| 31-40 | BCODE (2) | An interpolation code used by BITURP to find the intersections of wing percent lines with the body surface (see subroutine WBXX). This process involves finding body sections at various values of x. If BCODE (2) = 1, linear interpolation is used to find y as a function of z; this is equivalent to saying that the body is polygonal in cross section. If BCODE (2) = 2 (and there are at least three meridian lines), biquadratic interpolation is used; this is equivalent to specifying a body of curved cross section (but see RESTRICTIONS, under BITURP). |
| 41-50 | EPRC | A tolerance applied to body radius and $z_c$ (z-coordinate of the body any value of x) is less than EPRC, the corresponding radius or $z_c$ is set to zero. EPRC is introduced to help offset round-off errors. |

Two binary tapes are written by TFLAT; their formats are described in Appendix D.

USAGE:

```
COMMON /LGEOMD/LGDEF(3,6)
LOGICAL  LGDEF
DIMENSION  DAT(2)
CALL  TFLAT(DAT, LI, LO, LA, LB, LD, NBODY, IOK)
```

| | | |
|---|---|---|
| Input: | DAT | = Date (alphameric). |
| | LI | = Input tape number. |
| | LO | = Output tape number. |
| | LA | = Binary tape number; this tape is read by TFLAT (see Appendix D). |
| | LB | = Binary tape number; three records are to be written on this tape, unless a wing-only case is processed (see Appendix D, section 3). |
| | LD | = Binary tape number; six records are to be written on this tape (see Appendix D, section 2). |
| | LGDEF | = Logical array, in COMMON, which indicates body and wing status. LGDEF(1, 3) = .TRUE. if a body definition was requested; LGDEF(3, 3) = .TRUE. if the body definition was successfully written on tape LA. LGDEF(1, 1) = .TRUE. if a wing definition was requested; LGDEF(3, 1) = .TRUE. if the wing definition was successfully written on tape LA. The other elements of LGDEF are not used by TFLAT. |
| Output: | NBODY | = Code to indicate a successful body-alone case; then NBODY = -1. Otherwise, NBODY = 0. |
| | IOK | = Error indicator, which is zero if success. |

SUBPROGRAMS CALLED:

WBXX
TRAPCT
BITURP
TFLAT1
TFLATM
TFLATW
TFLATX
TRAV
FSR

ERROR RETURNS: If an error is detected, a message is written on the output tape. Then data cards on the input tape are skipped over until the next case, or end of data, is found. A new case is signaled by the word DEFINE or DEF1NE in columns 1-6 of a data card; end of data is signaled by the word END OF. The input tape is then backspaced one record.

The argument IOK is set to one of the following values if an error is found:

| IOK | Explanation |
|---|---|
| 2 | A body or wing definition failed in a previous link. |
| 4 | Error reading body stations from tape LA. Either the number of stations is less than 2 or greater than 51, or something is wrong with the tape. |
| 6 | Error reading body meridian lines from tape LA. Something is wrong with the tape or record 4 has more than 4368 words. |
| 8 | Error from subroutine TFLAT1. |
| 10 | Error reading wing percent lines from tape LA. Record 7 is not correct. |
| 12 | Record 7 on tape LA indicates that record 8 has more than 4368 words. |
| 14 | Error from subroutine WBXX in finding wing percent line intersections with the body (before transformation). |
| 16 | Either the leading or trailing edge of the wing does not intersect the body. |
| 24 | Error from subroutine WBXX in finding the intersections of the flattened wing with the transformed body. |
| 26 | Error from subroutine TFLATX. One or more of the flattened wing percent lines did not intersect the transformed body. |
| 32 | Error from subroutine BITURP in interpolating points on the radius versus station or z-centroid versus station curve. |

If IOK = 2, the message "ERROR IN PREVIOUS LINK" is written. If IOK > 2, the message:

"TFLAT ERROR i, CODE j
IREC = $i_1$ $i_2$ $i_3$ $i_4$"

is written, where i = IOK, j = error code from a subroutine (if IOK = 8, 14, 24, 26, or 32), and $i_1$, $i_2$, $i_3$, $i_4$ are the first four elements of the last odd-numbered record read from tape LA. If IOK = 14 or 16, the transformed wing percent lines are written by subroutine TFLATW to aid in diagnosing the error.

RESTRICTIONS: Only a single wing surface can be processed. If tape LA is written by the geometry definition program, a lower-wing case cannot be handled since TFLAT only examines the upper-wing portion of tape LA. The number of words in records 4 and 8 cannot exceed 4368.

STORAGE: $11957_{10} = 27265_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TFLAT1 |
| PURPOSE: | To transform body meridian lines to a new coordinate system, and to find the centroid and equivalent body radius at a set of body stations. |
| METHOD: | This subroutine is used by subroutine TFLAT. The body centroid at each end of the body is found by subroutine BODCR. Then a new coordinate system is set up such that the origin is at the forward end centroid and the new x-axis passes through the aft-end centroid. After transforming the given body to the new coordinate system, the body radius and z-coordinate of the centroid is found at each given body station. |
| USAGE: | DIMENSION STA(51), B(2448), S(100, 2), E(2000), RT(3, 4), ALFBET(2), RZ(51, 2) NU(3) |

CALL TFLAT1 (CHD, NSTA, STA, NBPCT, B, S, E, RT, ALFBET, RZ, EPRC, NU)

Input:

| | | |
|---|---|---|
| CHD | = | A chord-arc tolerance, read from data card by subroutine TFLAT. |
| NSTA | = | Number of body stations. |
| STA | = | Array of original body stations (see Output, below). |
| NBPCT | = | Number of body meridian lines. |
| B | = | Array of points on original body meridian lines (see Output, below). This array (with a header) has a format as described in Appendix D, |
| EPRC | = | A tolerance, read from data card by subroutine TFLAT. |
| NU | = | Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, Then if NU(3) > 0 and NU(2) > 0, an error message is written. |

| | | |
|---|---|---|
| Output: STA | = | Array of transformed body stations. |
| B | = | Array of points on transformed body meridian lines (with header). |
| S | = | Scratch array. |
| E | = | Scratch array. |
| RT | = | Transformation matrix (see TROTPT). |
| ALFBET | = | Transformation constants to convert original body stations to new stations (see TFLAT). |
| RZ | = | Array of body radii and z-centroid corresponding to each body station. |
| NU | = | Error indicator array (see Input) NU(1) = 0 for success. |

SUBPROGRAMS CALLED:

UVECN
BSCALE
BODCR
VCROS
TRAPCT
TRAV

ERROR RETURNS: If an error is detected, the message "TFLAT ERROR i, CODE J, ...." is written.

| i | j | Explanation |
|---|---|---|
| 1 | k | Error k from BODCR at forward body end. |
| 2 | k | Error k from BODCR at aft body end. |
| 3 | 0 | The original body length is zero. |
| 4 | k | Error k from BSCALE. |
| 5 | 0 | The transformed body length is zero. |
| 6 | k | Error k from BODCR at some transformed body station (an additional comment will identify the station). |

RESTRICTIONS: It is assumed that the given body meridian lines describe only half of a body that is symmetrical about a plane parallel to the xz plane.

STORAGE: $381_{10} = 575_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TFLATM |
| PURPOSE: | To write body meridian line points on an output tape. |
| METHOD: | This is primarily an output routine for TFLAT. The meridian lines are given in a single array with a format such as that described in Appendix D. The polar coordinates of each point are computed in a plane normal to the x-axis, with polar origin at the body axis. |
| USAGE: | DIMENSION B(1), AXIS(2), TITLE(12), DAYT(2), STA(1) |
| | CALL TFLATM (B, NB, AXIS, TITLE, DAYT, LO, STA) |
| | Input: B = Array of meridian lines, with header. |
| | NB = Number of meridian lines. |
| | AXIS = y, z coordinates of main body axis. |
| | TITLE = Title (alphameric). |
| | DAYT = Date (alphameric). |
| | LO = Output tape number. |
| | STA = Not used. |
| | Output: = Print-out on tape LO |
| SUBPROGRAMS CALLED: | ATN1<br>SQRT (Built-in function)<br>TRAV |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $346_{10} = 532_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TFLATW |
| PURPOSE: | To write wing percent lines on an output tape. |
| METHOD: | This is an output subroutine used by subroutine TFLAT. The percent lines are given in a single array with a format as described in Appendix D. |
| USAGE: | DIMENSION B(1), TITLE(12), DAYT(2) |
| | CALL TFLATW (B, NB, TITLE, DAYT, LO) |
| | Input: B = Array of percent lines, with header. |
| | NB = Number of percent lines. |
| | TITLE = Title (alphameric). |
| | DAYT = Date (alphameric). |
| | LO = Output tape number. |
| SUBPROGRAM CALLED: | TRAV |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $312_{10} = 470_8$ |

| | |
|---|---|
| **SUBJECT:** | FORTRAN IV Subroutine TFLATX |
| **PURPOSE:** | To write wing-body intersection points on an output tape. |
| **METHOD:** | This is an output subroutine for subroutine TFLAT (see subroutine TFLAT). The polar coordinates of each point are computed and written on the output tape along with the x, y, z coordinates. |

**USAGE:**

DIMENSION DAT(2), W(1), WB(3,NW)

CALL TFLATX (LO, DAT, NW, W, WB, NU)

Input:
- LO = Output tape number.
- DAT = Date (alphameric).
- NW = Number of wing percent lines (also number of wing-body intersection points).
- W = Array of wing percent lines, with header (see Appendix D).
- WB = Array of points ($x_i$, $y_i$, $z_i$) that are the intersections of wing percent lines with body surface.

Output:
- NU = Error indicator, which is zero for success.

**SUBPROGRAMS CALLED:**

ATN1
SQRT (Built-in function)
TRAV

**ERROR RETURNS:** NU = 1 if any intersection point is not valid (indicated by coordinates all $10^{30}$).

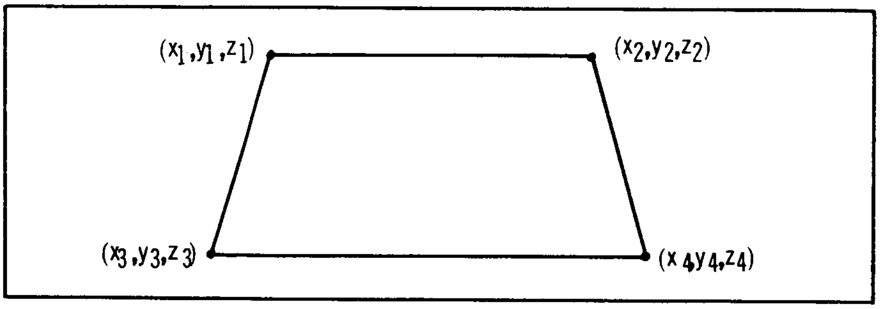
**RESTRICTIONS:** None

**STORAGE:** $240_{10} = 360_8$

SUBJECT: FORTRAN IV Subroutine THETAB

PURPOSE: To calculate body panel theta-inclination angles.

METHOD: The program calculates the panel theta-inclination angle by use of standard geometric formulae. The inclination angle $\theta$ is the angle of the panel in the x, y, z coordinate system of the body as viewed down the x-axis. A program check determines if the panel is "horizontal" by comparing the z-coordinates of the two trailing-edge panel corner points. If the panel is horizontal, theta is set equal to zero. For a nonhorizontal panel defined as follows,




theta is calculated from the formula,

$$\theta = \frac{|z_4 - z_3|}{(z_4 - z_3)} \cos^{-1}\left[\frac{A}{(A^2 + B^2)^{1/2}}\right]$$

for which,

$$A = \begin{vmatrix} x_1 & y_1 & 1 \\ x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \end{vmatrix}$$

$$B = - \begin{vmatrix} x_1 & z_1 & 1 \\ x_3 & z_3 & 1 \\ x_4 & z_4 & 1 \end{vmatrix}$$

USAGE: COMMON (See subroutine OPCAMI for unlabeled COMMON description).

COMMON /COM1/ (See subroutine PANEL)

Input: NPER1
NPLN1

COMMON /COM2/ (See subroutine BODY)

Input: XCOR
YCOR
ZCOR

Output: THETA

CALL THETAB

SUBPROGRAMS CALLED: ACOS
SQRT (Built-in function)

ERROR RETURNS: None

STORAGE: $244_{10} = 364_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine TRAPCT |
| PURPOSE: | To perform a coordinate transformation on a set of body meridian lines or wing percent lines. |
| METHOD: | Subroutine TROTPT is applied to transform each point, using a given transformation matrix. |
| USAGE: | DIMENSION B(1), RT(3, 4) |
| | CALL TRAPCT (B, N, RT) |
| | Input: B = Meridian or percent line array (see subroutines TFLAT and TFLAT1). |
| | N = Number of meridian or percent lines. |
| | RT = Transformation matrix (see subroutine TROTPT). |
| | Output: B = Transformed array. |
| SUBPROGRAM CALLED: | TROTPT |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $86_{10} = 126_8$ |

**SUBJECT:** FORTRAN Subroutine TRAV

**PURPOSE:** To move a block of N cells starting at location B to another region starting at location A.

**METHOD:**

```
        DO   100    I = 1, N
100     A(I)  =  B(I)
```

**USAGE**

CALL TRAV (A, B, N)
DIMENSION A(N), B(N)

**SUBPROGRAM CALLED:** None

**ERROR RETURNS:** None

**RESTRICTIONS:** $N > 0$

**STORAGE:** $35_{10} = 43_8$

**SUBJECT:** FORTRAN IV Subroutine TROTPT

**PURPOSE:** Given the coordinates of a point (in 3-space) in one of two rectangular Cartesian coordinate systems and the rotation and translation matrices relating the systems, to compute the coordinates of the point in the other system.

**METHOD:** Let an arbitrary point have the coordinates $P = (X_1, Y_1, Z_1)$ in an $X_1 Y_1 Z_1$ coordinate system, and the coordinates $Q = (X_2, Y_2, Z_2)$ in an $X_2 Y_2 Z_2$ coordinate system. Let R be a 3-by-3 rotation matrix constructed as follows: the first column is a unit vector, in the $X_1 Y_1 Z_1$ system, along the $+X_2$ axis; similarly, the second column is a unit vector along the $+Y_2$ axis, and the third column is a unit vector along the $+Z_2$ axis. Let T be a 1-by-3 translation matrix that contains the $X_1 Y_1 Z_1$ coordinates of the origin of the $X_2 Y_2 Z_2$ system. Then, if P and Q are considered to be 1-by-3 matrices,

$$Q = (P - T)R$$

$$P = QR^{-1} + T$$

**USAGE:**

DIMENSION R(3 3), T(3), P(3), Q(3)

CALL TROTPT (R, T, K, P, Q)

Input: R = Rotation matrix.

T = Translation matrix.

K = A code that is 1 if Q is input and P is output, or 2 if P is input and Q is output.

P or Q = Point coordinates.

Output: Q or P = Point coordinates.

**SUBPROGRAM CALLED:** None

**ERROR RETURNS:** None

**RESTRICTIONS:** The rotation matrix must be orthogonal. The same array may be used for both P and Q; e.g., CALL TROTPT(R, T, 1, Q, Q).

**STORAGE:** $112_{10} = 160_8$

**SUBJECT:** FORTRAN IV Subroutine TTAPE

**PURPOSE:** To copy matrices from tape to tape.

**METHOD:** A single matrix (or multiple matrices) is read from one tape and written on a second tape under binary format.

**USAGE:** CALL TTAPE (KIND, IN, JN, NTIN, NTOUT, A, B, C)

DIMENSION A(IN), B(IN), C(IN)

Input:

| | | |
|---|---|---|
| KIND | = | Code for number of matrices to be copied. If KIND = 0, one matrix (A) to be copied. If KIND $\neq$ 0, three matrices (A, B, and C) to be copied. |
| IN | = | Number of rows. |
| JN | = | Number of columns. |
| NTIN | = | Tape from which matrices are read. |
| NTOUT | = | Tape on which matrices are written. |
| A, B, C | = | Dummy arrays. |

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $159_{10} = 237_8$

SUBJECT: FORTRAN IV Subroutine TVEL

PURPOSE: To compute the velocity components resulting from a specified wing thickness distribution from the velocity components resulting from a wing thickness distribution of unit incidence.

METHOD: The velocity components in the x, y, and z directions on panel i resulting from a specified wing thickness distribution of incidence $\alpha_{T_j}$ are found by:

$$U_i = \sum_{j=1}^{NWING} U_{ij} \cdot \alpha_{T_j}$$

$$V_i = \sum_{j=1}^{NWING} V_{ij} \cdot \alpha_{T_j}$$

$$W_i = \sum_{j=1}^{NWING} W_{ij} \cdot \alpha_{T_j}$$

where $U_i$, $V_i$ and $W_i$ = The $i^{th}$ velocity components resulting from a wing thickness distribution of unit incidence.

NWING = The total number of wing panels.

The normal component of velocity on a body panel i resulting from wing thickness is given by

$$n_{B_i} = \sum_{j=1}^{NWING} A_{BWS_{ij}} \cdot \alpha_{T_j}$$

where $A_{BWS}$ = The normal velocity on the body resulting from wing thickness distribution of unit incidence.

USAGE: CALL TVEL (A, B, C, D, ALPHAT, UBWT, VBWT, WBWT, UWWT, VWWT, WWWT, AN, CHORD, THICK, NROW)

COMMON DATE(2), NTAPEA, NTAPEB, NTAPEC, NTAPED, NTAPEE, NTAPEF, NTAPEI, NTAPEO, NBODY, NWING, XMACH, SYM, KACE

DIMENSION A(210), B(210), C(210), D(210),

ALPHAT (110), UBWT (100), VBWT (100), WBWT (100),

UWWT (110), VWWT (110), WWWT (110), AN (100),

CHORD (210), NROW (2)

Input:

| | | |
|---|---|---|
| A, B, C, D | = | Dummy arrays |
| ALPHAT | = | Array of wing thickness slopes. |
| CHORD | = | Array of panel chord lengths. |
| NROW | = | Array of number of rows of panels. For wing-alone configurations, NROW (1) = number of panel rows on wing and NROW (2) is dummy argument. For wing-body configurations, NROW (1) = number of panel rows on body and NROW (2) = number of panel rows on wing. |
| THICK | = | Code. If THICK = 0, wing thickness effects to be ignored. If THICK = 1, wing thickness effects to be included. |

Output:

| | | |
|---|---|---|
| UBWT, VBWT, WBWT | = | Arrays of velocity components in x-, y- and z-directions on the body due to a given wing thickness distribution. |
| UWWT, VWWT, WWWT | = | Arrays of velocity components in x-, y- and z-directions on the wing due to a given wing thickness distribution. |

| | | |
|---|---|---|
| | AN | = Array of normal velocity components on the body due to a given wing thickness distribution. |

| | |
|---|---|
| SUBPROGRAM CALLED: | FSF |
| ERROR RETURNS: | None |
| RESTRICTIONS: | NBODY $\leq$ 100 |
| | NWING $\leq$ 110 |
| STORAGE: | $580_{10} = 1104_8$ |

| | |
|---|---|
| SUBJECT: | FORTRAN IV Subroutine UNLOAD |
| PURPOSE: | To rewind and unload a logical tape unit. The unload operation prevents any further reading or writing on the specified unit until the unit is readied by an operator. |
| METHOD: | The program delays until all I-O operations are finished, executes a rewind and unload command, then delays until the channel is again out of operation. |
| USAGE: | CALL UNLOAD (NT)<br><br>Input: NT = logical tape number |
| SUBPROGRAMS CALLED: | None |
| RESTRICTIONS: | None |
| ERROR RETURNS: | None |
| STORAGE: | $43_{10} = 53_8$ |

SUBJECT: FORTRAN IV Subroutine USETAP

PURPOSE: To read the aerodynamic matrices for a given configuration and Mach number from a previously saved tape and to rewrite them on other logical tapes in the formats required for a computer run.

METHOD: The previously saved tape is mounted on logical tape NTAPEC and the aerodynamic matrices are read and rewritten on other logical tapes as indicated below.

For wing alone configurations, the matrices are written in the order:

| | New Tape | Matrix | Description |
|---|---|---|---|
| 1. | NTAPEA | $\left[A_{WW}\right]$ | Aerodynamic influence coefficients due to wing sources. |
| 2. | NTAPEA | $\left[WW\right]$ | Drag minimization. |
| 3. | NTAPEA | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift. |
| 4. | NTAPEA | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift and pitching moment. |
| 5. | NTAPEB | $\left[U_{WW}\right]$ | Velocity components due to wing sources. |
| | | $\left[V_{WW}\right]$ | |
| | | $\left[W_{WW}\right]$ | |
| 6. | NTAPEB | $\left[U_{WW}\right]$ | Velocity components due to wing surface vortices. |
| | | $\left[V_{WW}\right]$ | |
| | | $\left[W_{WW}\right]$ | |
| 7. | NTAPEE | $\left[A_{WW}\right]$ | Aerodynamic influence coefficients due to wing surface vortices. |

| | | | |
|---|---|---|---|
| 8. | NTAPEE | $\left[A_{WW}\right]^{-1}$ | Inverse of matrix of aerodynamic influence coefficients due to wing surface vortices. |

For wing-body configurations, the matrices are written in the following order:

| | New Tape | Matrix | Description |
|---|---|---|---|
| 1. | through 5. | (same as wing-alone configurations) | |
| 6. | NTAPED | $\left[U_{BB}\right]$ $\left[V_{BB}\right]$ $\left[W_{BB}\right]$ | Velocity components on the body due to body surface vortices. |
| 7. | NTAPED | $\left[U_{WB}\right]$ $\left[V_{WB}\right]$ $\left[W_{WB}\right]$ | Velocity components on the wing due to body surface vortices. |
| 8. | NTAPED | $\left[U_{BW}\right]$ $\left[V_{BW}\right]$ $\left[W_{BW}\right]$ | Velocity components on the body due to wing surface vortices. |
| 9. | NTAPED | $\left[U_{WW}\right]$ $\left[V_{WW}\right]$ $\left[W_{WW}\right]$ | Velocity components on the wing due to wing surface vortices. |
| 10. | NTAPEE | $\left[A_{R}\right]$ | "Reduced" aerodynamic matrix. |
| 11. | NTAPEE | $\left[A_{R}\right]^{-1}$ | Inverse of "reduced" aerodynamic matrix. |
| 12. | NTAPEF | $\left[A_{BB}\right]^{-1}$ | Inverse of matrix of aerodynamic influence coefficients. |

| | New Tape | Matrix | Description |
|---|---|---|---|
| 13. | NTAPEF | [D] | Product matrix. |

$$[D] = \left[A_{WB}\right] \cdot \left[A_{BB}\right]^{-1}$$

| | | | |
|---|---|---|---|
| 14. | NTAPEF | [E] | Product matrix. |

$$[E] = \left[A_{BB}\right] \cdot \left[A_{BW}\right]$$

This subroutine is not used for body-alone configurations.

USAGE: CALL USETAP

Input:
NTAPEA, NTAPEB, NTAPED, NTAPEE, NTAPEF = Logical tape numbers on which aerodynamic matrices are rewritten.
NTAPEC = Logical tape number of save tape and from which aerodynamic matrices are read.

SUBPROGRAMS CALLED:
TTAPE
FSF

ERROR RETURNS: None

RESTRICTIONS: This subroutine is not used for body-alone configurations.

STORAGE: $1309_{10} = 2435_8$

**SUBJECT:** FORTRAN Function UVECN

**PURPOSE:** Given two points A and B in N-space, to find components of vector (or unit vector) directed from A to B and distance between A and B.

**METHOD:** Vector $\overline{V} = \overline{B} - \overline{A}$ and sum $S = V_1^2 + V_2^2 + \ldots + V_n^2$ are formed. If $S > 0$, then distance $D = \sqrt{S}$; if a given code indicates that a unit vector is required, then $\overline{V} = \overline{V}/D$. Function value UVECN is set to D before return. If $S = 0$, then $\overline{V} = \overline{0}$ and $D = 0$.

**USAGE:**

D = UVECN (A, B, V, K, N)
DIMENSION A(N), B(N), V(N)

Input:
- A, B = Point coordinates
- K = If K = 0, then a unit vector $V = (B - \overline{A})/|(\overline{B} - \overline{A})|$ is desired;
  If $K \neq 0$, then $\overline{V} = \overline{B} - \overline{A}$.
- N = Dimension of the space in which A and B are defined.

Output:
- V = Vector $\overline{V}$
- D = Distance from A to B

SUBPROGRAM CALLED: SQRT (Built-in function)

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $80_{10} = 120_8$

SUBJECT: FORTRAN IV Subroutine VCROS

PURPOSE: To find cross-product of two vectors.

METHOD: Given vectors $\bar{A} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $\bar{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, then cross-product, A x B, is vector

$$\bar{V} = \begin{pmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{pmatrix}$$

USAGE:

```
DIMENSION  A(3), B(3), V(3)
CALL VCROS (A, B, V, D, K)
```

Input: $A = \bar{A}$

$B = \bar{B}$

$K = 0$ if $\bar{V}$ is not to be normalized

$\neq 0$ if $\bar{V}$ is to be normalized

Output: $V = \bar{A} \times \bar{B}$ if K was 0

$= (\bar{A} \times \bar{B})/D$ if K was not 0 and $D > 0$.

D = Magnitude of V (before normalization)

$K = 0$ if success

$= 1$ if $K \neq 0$ on input and $D = 0$.

SUBPROGRAM CALLED: VDOTP

SQRT (Built-in function)

RESTRICTIONS: Note that K is both an input and an output.

STORAGE: $106_{10} = 152_8$

SUBJECT: FORTRAN Function VDOTN

PURPOSE: To find dot-product of two vectors in N-space.

METHOD: Standard

USAGE: DP = VDOTN (A, B, N)

DIMENSION A(N), B(N)

Input: A = Vector of dimension N

B = Vector of dimension N

Output: $DP = \overline{A} \cdot \overline{B} = \sum_{i=1}^{N} A_i B_i$

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: $N > 0$

STORAGE: $40_{10} = 50_8$

| | |
|---|---|
| SUBJECT: | FORTRAN IV Function VDOTP |
| PURPOSE: | To find dot-product of two vectors. |
| METHOD: | Given two vectors $\overline{A} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $\overline{B} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, dot-product is the scalar $P = a_1 b_1 + a_2 b_2 + a_3 b_3$. |
| USAGE: | DIMENSION A(3), B(3)<br>P = VDOTP (A, B)<br>Input: A = Vector $\overline{A}$<br>B = Vector $\overline{B}$<br>Output: $P = A \cdot B$ |
| SUBPROGRAM CALLED: | None |
| ERROR RETURNS: | None |
| RESTRICTIONS: | None |
| STORAGE: | $43_{10} = 53_8$ |

SUBJECT: FORTRAN IV Subroutine VEL1

PURPOSE: To compute velocity components and pressures induced at field points by sources and doublets on an isolated body (Refer to KARMOR).

METHOD: The equations used in this routine are given in Part I, section 4.4.

USAGE:

```
COMMON/FLOV1/KACE, NPANEL, NBODY, NWING,
NBODYS, NWINGS, NROW(2), XMACH, SYM

DIMENSION A(210, 3, 4), B(210, 3, 4), C(210, 3, 4),
NPART(210)

DIMENSION ALPHAS (210), THETA(210)

DIMENSION XB(50), R(50), WT(120)

DIMENSION T(50), TC(50), SST(210), CHORD(210)

COMMON/FLOV2/A, B, C, NPART, ALPHAS, THETA,
XB, R, WT, T, TC, SST, CHORD

COMMON/FLOV3/T11, TC11

COMMON/THICK/THKW, ARA

CALL VEL1 (X, Y, Z, UU, VV, WW)
```

Input:

| | | |
|---|---|---|
| X | = | x-coordinate of field point. |
| Y | = | y-coordinate of field point. |
| Z | = | z-coordinate of field point. |
| XB | = | Array of x-coordinates of the body line source and doublet segments. |
| R | = | Array of body radii defined at the XB coordinates. |
| T | = | Array of line source strength. |
| TC | = | Array of doublet strengths. |
| T11 | = | Strength of linearly varying source. |
| TC11 | = | Strength of linearly varying doublet. |
| ARA | = | Angle of attack. |

| | | | |
|---|---|---|---|
| | Output: | UU | = x-direction velocity component due to body. |
| | | VV | = y-direction velocity component due to body. |
| | | WW | = z-direction velocity component due to body. |

SUBPROGRAMS CALLED: ALOG, COS, SQRT, SIN, ATN1 $\Big\}$ (Built-in functions)

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $400_{10} = 740_8$

SUBJECT: FORTRAN IV Subroutine WBX

PURPOSE: To find the intersections of a set of wing percent chord lines with a body surface and to delete the portions of percent chord lines inside the body.

METHOD: The body surface is defined by an array of NB meridian lines and a code that determines body shape between meridian lines (see subroutine WBXUL). Each wing percent chord line consists of a series of straight-line segments connecting three-dimensional points. The intersection of each wing percent chord line with the body surface is then calculated.

The body is cut (see subroutine WBXC) by a transverse plane that passes through the first wing percent chord point. The horizontal distance normal to the x-axis from the percent point to the body surface is found in this region. This calculation is repeated for successive percent chord line points until a change in sign of the distance is detected; this means that the last two wing points, $W_1$ and $W_2$ for example, are on opposite sides of the body surface. An iterative procedure is now used to locate the intersection point.

Let $\alpha$ be the distance along the line segment $\overline{W_1 W_2}$ from $W_1$ to some point W on the segment. Let $f(\alpha)$ be the horizontal distance from W to the body surface. If L is the length of $\overline{W_1 W_2}$, then f(0) and f(L) have opposite signs (because f(0) corresponds to $W_1$ and f(L) corresponds to $W_2$). Subroutine FROOTA is used to iterate on $\alpha$ until $f(\alpha)$ is as small as possible. The point W is then on or very near the body surface; the corresponding point on the body surface is taken as the intersection.

If the first wing percent chord line point is forward or aft of the body, the transverse plane through the point does not intersect the body; WBX then continues to process points until a body intersection is found. If all points so far processed are on the same side of the body surface, and the current point is forward or aft of the body, then the previous point is taken as $W_1$; $W_2$ is found by subroutine WBXD as the intersection of the percent chord line with a transverse plane through the appropriate body end. If all percent chord lines are processed without finding two adjacent points on opposite sides of the body surface, the intersection point is taken as $(10^{30}, 10^{30}, 10^{30})$ to indicate that no intersection point was found.

USAGE:

DIMENSION B(1), W(1), S(2, NB), BW(3, NW), NU(3)

CALL WBX (B, NB, W, NW, EPS, KODE, S, BW, NU)

Input:

- B = Array of body meridian lines (described in subroutine WBXUL).
- NB = Number of meridian lines.
- W = Array of wing percent chord lines (described in subroutine WBXUL).
- NW = Number of percent chord lines.
- EPS = Tolerance — a message will be written on the output tape (see NU) if the error in finding an intersection is greater than EPS.
- KODE = A code that determines the shape of the body surface between meridian lines (see subroutine WBXUL).
- NU = Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Scratch:

- S = Storage for sections through the body.

Output:

- W = Array of wing percent chord lines, truncated at the body surface.
- BW = Array of intersection points ($x_i$, $y_i$, $z_i$).
- NU = Error indicator array: NU(1) is zero if no error is detected. NU(3) may have been changed (see INPUT, above).

SUBPROGRAMS CALLED:

UVECN
FROOTA
TRAV
WBXA
WBXC
WBXD

ERROR RETURNS: The following message is written (see NU) if an error is detected: "ERROR i, CODE j IN SUBROUTINE WBX, WING PERCENT LINE m, POINT n."

| <u>i</u> | <u>j</u> | <u>Explanation</u> |
|---|---|---|
| 4 | 1 | WBXA error; see note 1, below. |
| 5 | 2 | WBXD error; see message written by WBXD. |
| 6 | -3 | WBXC error; first and last points in a body section (through a wing percent line point) have the same z-coordinate. |
| 7 | 3 | WBXC error; two points in body section, near the wing-body intersection, have the same z-coordinate |
| 8 | 1<br>2 | A wing percent chord line point is above or below the body section and has the same y-coordinate as the first or last section point. |
| 9 | - | See note 1. |
| 10 | k | WBXC error k in iteration loop. |
| 11 | k | FROOTA error; see note 1. |
| 12 | 0 | Iteration to find intersection did not converge; see note 1. |

Note 1: Machine or program error.

RESTRICTIONS: If a wing percent chord line intersects the body surface more than once, only one intersection point will be found. Body sections in the wing-body intersection region must be single-valued in y as a function of z. Wing percent chord lines must have at least one point that lies between body station planes through the body ends. A percent chord line intersection will not be found, even though it exists, if all percent chord points except the last are forward (or aft) of the body.

STORAGE: $605_{10} = 1135_8$

SUBJECT: FORTRAN IV Subroutine WBXA

PURPOSE: To modify a wing percent chord line array so that the first part of a given percent chord line is removed.

METHOD: The format of the wing percent chord line array, B, is explained in Appendix D. K, the number of points to be removed, and P, the new first-point, are given. If K = 0, P is not used and the entire old percent chord line is moved to its new position. Otherwise, K points are removed and P, followed by the remaining points, is moved. The corresponding portion of the array header is adjusted to reflect the new location of the percent chord line in the array and the new number of points. The remaining percent chord lines are not disturbed.

USAGE: DIMENSION B(1), P(3)

CALL WBXA (B, I, K, P, NU)

| | | |
|---|---|---|
| Input: | B | = Percent chord line array. |
| | I | = The percent chord line to be modified. |
| | K | = Number of points to be removed. |
| | P | = New first-point. |
| Output: | B | = Modified percent chord line array. |
| | NU | = Error indicator, which is zero if successful. |

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = 1 if $K < 0$ or if $K \geq$ number of points in old percent chord line.

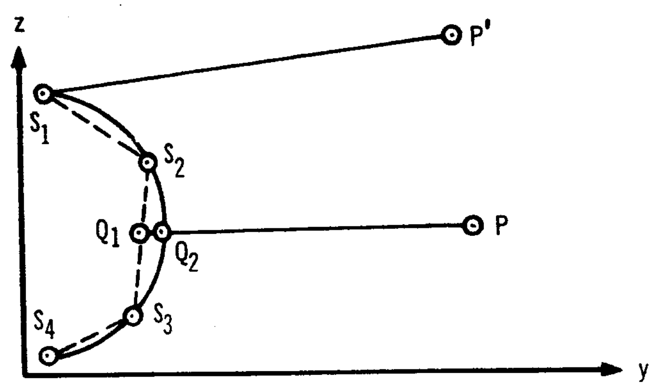
RESTRICTIONS: WBXA must be called successively with I = 1, 2, ..., N where N = number of percent chord lines.

STORAGE: $202_{10} = 312_8$

**SUBJECT:** FORTRAN IV Subroutine WBXC

**PURPOSE:** To find the distance from an arbitrary point to a body surface.

**METHOD:** The body is described by NB meridian lines (see subroutine WBXUL). Let $P(x_P, y_P, z_P)$ be the given point. Subroutine BCUTX is used to find the intersection points $(y_i, z_i)$ of the plane $x = x_P$ with the meridian lines. The points form a body section, S.




Let $Q(x_Q, y_Q, z_Q)$ be a point on the body so that $x_Q = x_P$, $z_Q = z_P$. The value of $y_Q$ is found by interpolation at $z = z_Q$ from the table $(z_i, y_i)$, using subroutine BITURP. If a given code is 1, the interpolation is linear ($Q_1$ in figure); if the code is 2, the interpolation is biquadratic ($Q_2$ in figure). This code effectively determines the body shape between meridian lines. If P should be above the body (see P' in drawing), then Q is taken as the first meridian line point. Similarly, the last meridian line point is used for Q if P is below the body. The required distance from P to the body surface is $y_P - y_Q$.

**USAGE:**

DIMENSION B(1), P(3), S(2, NB), Q(3)

CALL WBXC (B, NB, EP, P, KODE, S, D, Q, NU)

Input: B = Array of body meridian line points, with header (see subroutine WBXUL).

| | | |
|---|---|---|
| | NB | = Number of body meridian lines. |
| | EP | = Tolerance used by BCUTX in finding intersections of body meridian lines with the plane x = P(1). |
| | P | = x, y, z coordinates of given point. |
| | KODE | = Code, used by BITURP, that determines the shape of the body between meridian lines. KOD = 1 if polygonal, 2 if not (see METHOD, above). |
| Output: | S | = Section points ($y_i$, $z_i$), i = 1, 2,.., NS, where NS $\leq$ NB. |
| | D | = Distance. |
| | Q | = Point on body surface. |
| | NU | = Error indicator array. NU = 0, 1 or 2 if successful; 0 if Q(3) = P(3), 1 if $S_1$ was used for Q, 2 if $S_{NB}$ was used for Q. |

SUBPROGRAMS CALLED: BITURP
BCUTX

ERROR RETURNS: NU = -5 if the number of section points is less than 2; this may be caused by NB < 2 or failure of the plane $x = x_P$ to cut some of the meridian lines. NU = -3 if $S_1$ and $S_{NB}$ have the same z-coordinate.

RESTRICTIONS: The value of D returned is the horizontal distance between P and Q; if P is above or below the body, the true distance is greater than D. If all meridian lines do not have the same x-range, the number of section points (S) may possibly be less than NB; see BCUTX.

STORAGE: $139_{10} = 213_8$

**SUBJECT:** FORTRAN IV Subroutine WBXD

**PURPOSE:** To find a point on a wing percent chord line which lies in the first or last body station plane, and to determine the horizontal distance of that point from the body surface.

**METHOD:** Subroutine WBXD is called by subroutine WBX only if the following situation arises: At least one point, $W_1$, for instance, on some wing percent chord line has been processed; but the next point, $W_2$, is either forward or aft of the body. Let BXF and BXL be the extreme x-coordinates of body meridian line 1; let WXF and WXL be the x-coordinates of $W_1$ and $W_2$, respectively. If BXF lies in the range (WXF, WXL), a plane x = BXF is formed; otherwise BXL is assumed to be the range and the plane is taken as x = BXL. The line segment $\overline{W_1W_2}$ is intersected with the plane to obtain point P, which lies on the wing percent line. The horizontal distance from P to the body, and the corresponding point Q on the body are then found by subroutine WBXC.

**USAGE:** DIMENSION R(4), B(1), P(3), Q(3), S(2, NB), W(3, 2), NU(3)

Call WBXD (R, B, NB, KODE, P, Q, S, W, NU)

Input:

- B = Array of body meridian line points, with header (see subroutine WBXUL).
- NB = Number of body meridian lines.
- KODE = Body shape code (see WBXC).
- W = Two adjacent points on a wing percent chord line (see $W_1$ and $W_2$, METHOD, above).
- NU = Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output:

- S = Section points $(Y_i, Z_i)$.
- R = R(1) = x-coordinate of P; R(2) = horizontal distance = P(2) - Q(2); R(3) = R(1); R(4)

| | | |
|---|---|---|
| P | = | x, y, z coordinates of point on wing percent chord line. |
| Q | = | Corresponding point on body surface. |
| NU | = | Error indicator array. NU(1) is zero if success. NU(3) may have changed if an error was detected (see Input). |

SUBPROGRAMS CALLED: POLXN
WBXC

ERROR RETURNS: A message "ERROR i, CODE j, IN SUBROUTINE WBXD" is written if an error is detected.

| i | j | Explanation |
|---|---|---|
| 1 | 0 | Machine or program error. |
| 2 | k | Error k from WBXC. If k = 1 or 2, point P is above or below the body and P(2) = Q(2). |

RESTRICTIONS: See WBXC

STORAGE: $219_{10} = 333_8$

**SUBJECT:** FORTRAN IV Subroutine WBXUL

**PURPOSE:** To find the intersections of upper and lower wing percent chord lines with a body surface, to delete the portions of percent chord lines that are inside the body, and to write results.

**METHOD:** A set of NB meridian lines defines the basic body surface. These lines may be visualized as stringers running lengthwise along the surface of the body. Each meridian line consists of a series of straight-line segments connecting three-dimensional points. The surface between meridian lines is determined by taking any section, normal to the X-axis, through the body; this yields the (Y, Z) coordinates of NB points on the surface. Under one option, the surface at this section is assumed to be polygonal (adjacent points connected by straight lines). In the other option, the surface between each pair of points is assumed to be represented by a cubic; points between meridian lines are interpolated by subroutine BITURP.

The wing surface (upper, lower, or both) is defined by NW wing percent chord lines, which are similar to meridian lines. The 0-percent line forms the leading edge of the wing, and the 100-percent line is the trailing edge. The wing surface is not defined between percent chord lines. The main function of WBXUL is to find the point of intersection of each wing percent chord line with the body surface.

All of the body meridian line points are stored in a single array. The first 3*NB cells form a "header" that is a key to the rest of the array. Cell 1 is a number that forms a label for the first meridian line. Cell 2 contains the number of points in the first meridian line. Cell 3 contains the cell number at which the x-coordinate of the first point in the first meridian line is located. Similarly, cells 4, 5, and 6 apply to the second meridian line, etc. This storage scheme, described in Appendix D, is also used for the upper wing percent lines and the lower percent lines.

The subroutine begins by a check that the body and at least one wing surface are defined. Then one card is read with FORMAT(2F10.0). The first field contains a code, later used by BITURP, that determines body shape between

meridian lines (1 if polygonal, 2 if not). The second field contains a tolerance, EPS, that is set to $10^{-5}$ if less than $10^{-5}$; subroutine WBX writes a message if the error in locating an intersection point is more than EPS. The following process is then carried out, for the upper wing surface and then the lower.

If the wing surface is defined, storage is reserved in the buffer for the intersection points (see Appendix A). Subroutine WBX is called to find the intersection points and remove that part of each wing percent chord line inside the body surface. Each wing percent chord line now starts at the body surface; this, of course, requires a new "header" for the array. The X, Y, Z and polar coordinates of each intersection point are written on the output tape. The polar coordinate system for each point lies in a plane through the point and normal to the X-axis, with origin at the known body axis; the angle is measured from the vertical (Z-axis). If any percent line fails to intersect the body surface, a message is written but the routine continues.

USAGE:

```
DIMENSION  B(1), DAT(2), LOK(3, 9), BTITLE(12),
WTITLE(12), BAXIS(2), NU(3)

LOGICAL  LGDEF(3, 6)

CALL WBXUL(B, LI, LO, DAT, LGDEF, LOK, BTITLE,
WTITLE, BAXIS, NU)
```

Input:

| | | |
|---|---|---|
| B | = | Storage buffer for variable length arrays. |
| LI | = | Input tape number. |
| LO | = | Output tape number. |
| DAT | = | Date (alphameric). |
| LGDEF(2, 1) | = | .TRUE. if upper wing is defined. |
| (2, 2) | = | .TRUE. if lower wing is defined. |
| (2, 3) | = | .TRUE. if body is defined. |
| LOK | = | Array that tells where in B the wing percent chord line arrays and body meridian line array are located, and the length of each array. Consider LOK(i, j): The upper wing percent chord line corresponds to |

LOK(i, 1), the lower to LOK(i, 2); the body corresponds to LOK(i, 3). An array starts in B(LOK(1, j)). LOK(2, j) = number of percent chord lines or meridian lines. LOK(3, j) = total number of elements in array.

**BTITLE** = Body title (alphameric).

**WTITLE** = Wing title (alphameric).

**BAXIS** = Y and Z coordinates of the body axis.

**NU** = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) =(NU(3) - 1). Then if NU(3) > 0 and NU(2)> 0, an error message is written.

Output: B = Storage buffer for variable-length arrays.

LGDEF(1, 5) = Set equal to LGDEF(1, 1) to indicate that a wing-body intersection was requested.

(1, 6) = Set equal to LGDEF(1, 2).

(2, 5) = Set to .TRUE. if intersections of the upper wing percent chord lines with the body were found.

(2, 6) = Set to .TRUE. if intersections of the lower wing percent chord lines with the body were found.

LOK = Array that tells where in B various arrays are stored (see Input). The truncated upper wing percent chord lines (part outside body) correspond to j = 3, the lower to j = 4; the upper intersection point array corresponds to j = 8, the lower to j = 9. An array starts in

= B(LOK(1, j)). LOK(3, j) = total number of elements in array. LOK(2, j) = number of percent lines if j = 3 or 4. LOK(2, j) is not used for j = 8 or 9.

NU = Error indicator array. NU(1) is zero if success. NU(3) may have been changed if an error was detected; see Input, above.

SUBPROGRAMS CALLED:

ATN1
WBX
MERR
SQRT (Built-in function)
TRAV
IRSERV, IPACK (see Appendix A)

ERROR RETURNS:

Function MERR is used to write a message "ERROR i, CODE j IN SUBROUTINE WBXUL DURING GEOMETRIC DEFINITION" if an error is detected.

| i | j | Explanation |
|---|---|---|
| 1 | 0 | Either the body or wing was not successfully defined. However, if both upper and lower wing surfaces were requested but one surface failed, this error will not stop WBXUL (provided the body is defined). |
| 2 | 0 | The storage buffer, B, is not large enough or, a storage error has been detected. |
| 3 | 0 | (same as above) |
| 4 | k | Error k from WBX. |

RESTRICTIONS:

The storage buffer, B, must have been initialized by subroutine INIBFR (see Appendix A) or its equivalent. The original wing percent lines (stored starting in B(LOK(1, i)), i = 1 and 2) are destroyed.

STORAGE:

$758_{10} = 1366_8$

**SUBJECT:** FORTRAN IV Subroutine WBXX

**PURPOSE:** To find the intersections of a set of wing percent chord lines with a body surface.

**METHOD:** The body surface is defined by an array of NB meridian lines and a code that determines body shape between meridian lines (see subroutine WBXUL). Each wing percent line consists of a series of straight-line segments connecting three-dimensional points. The intersection of each wing percent chord line with the body surface is found in turn.

The body is cut (see subroutine WBXC) by a transverse plane that passes through the first wing percent chord line point. In this section, normal to the X-axis, the horizontal distance (with sign) from the percent point to the body surface is found. This process is repeated for successive percent chord line points until a change in sign of the distance is detected; this means that the last two wing points, $W_1$ and $W_2$, for example, are on opposite sides of the body surface. An iterative procedure is now used to locate the intersection point.

Let $\alpha$ be the distance along the line segment $\overline{W_1W_2}$ from $W_1$ to some point W on the segment. Let $f(\alpha)$ be the horizontal distance from W to the body surface. If L is the length of $\overline{W_1W_2}$, then $f(0)$ and $f(L)$ have opposite signs (because $f(0)$ corresponds to $W_1$ and $f(L)$ corresponds to $W_2$). Subroutine FROOTA is used to iterate on $\alpha$ until $f(\alpha)$ is as small as possible. The point W is then on or very near the body surface; the corresponding point on the body surface is taken as the intersection.

If the first wing percent chord line point is forward or aft of the body, the transverse plane through the point does not intersect the body; WBXX then continues to process points until a body intersection is found. If all points so far processed are on the same side of the body surface, and the current point is forward or aft of the body, then the previous point is taken as $W_1$; $W_2$ is found by subroutine WBXD as the intersection of the percent chord line with a body station plane through the appropriate body end. If all percent chord lines are processed without finding two adjacent points on opposite sides of the body surface, the intersection point is taken as $(10^{30}, 10^{30}, 10^{30})$ to indicate that no intersection point was found.

USAGE:

DIMENSION B(1), W(1), S(2, NB), BW(3, NW), NU(3)

CALL WBXX (B, NB, W, NW, EPS, KODE, S, BW, NU)

Input:

| | | |
|---|---|---|
| B | = | Array of body meridian lines (described in subroutine WBXUL). |
| NB | = | Number of meridian lines. |
| W | = | Array of wing percent chord lines (described in subroutine WBXUL). |
| NW | = | Number of percent chord lines. |
| EPS | = | Tolerance; a message will be written on the output tape (see NU) if the error in finding an intersection is greater then EPS. |
| KODE | = | A code that determines the shape of the body surface between meridian lines (see subroutine WBXUL). |
| NU | = | Error indicator array. NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written. |

Scratch:

| | | |
|---|---|---|
| S | = | Storage for sections through the body. |

Output:

| | | |
|---|---|---|
| BW | = | Array of intersection points ($x_i$, $y_i$, $z_i$). |
| NU | = | Error indicator array. NU(1) is zero if no error is detected. NU(3) may have been changed (see Input, above). |

SUBPROGRAMS CALLED:

UVECN
FROOTA
TRAV
WBXC
WBXD

ERROR RETURNS:

The following message is written (see NU) if an error is detected: "ERROR i, CODE j IN SUBROUTINE WBXX, WING PERCENT LINE m, POINT n."

| i | j | Explanation |
|---|---|---|
| 5 | 2 | WBXD error; see message written by WBXD. |
| 6 | -3 | WBXC error; first and last points in a body section (through a wing percent line point) have the same z-coordinate. |
| 7 | 3 | WBXC error; two points in body section, near the wing-body intersection, have the same Z-coordinate |
| 8 | 1<br>2 | A wing percent line point is above or below the body section and has the same Y-coordinate as the first or last section point. |
| 9 | - | See note 1, below. |
| 10 | k | WBXC error k in iteration loop. |
| 11 | k | FROOTA error; see note 1. |
| 12 | 0 | Iteration to find intersection did not converge; see note 1. |

Note 1: Machine error or program error.

RESTRICTIONS: If a wing percent chord line intersects the body surface more than once, only one intersection point will be found. Body sections in the wing-body intersection region must be single-valued in y as a function of z. Wing percent chord lines must have at least one point that lies between body station planes through the body ends. A percent chord line intersection will not be found, even though it exists, if all percent points except the last are forward (or aft) of the body.

STORAGE: $584_{10} = 1110_8$

SUBJECT: FORTRAN IV Subroutine WING

PURPOSE: To serve as control program for wing paneling subroutines.

METHOD: Program uses a series of FORTRAN IV language "CALL" instructions to call required wing paneling subroutines.

The program has the following labeled COMMON statement that is also in all lower level wing geometry paneling subroutines,

```
COMMON /COM2/ IJ, NPTS(16), X(90, 16), Y(90, 16),
Z(90, 16), KPNT, VALUE(5), YCEPT(16), SLOPE,
KPANEL(2, 15), XCOR(16, 16), YCOR(16, 16), ZCOR(16, 16)
XINT(2, 15), YINT(2, 15), ZINT(2, 15), XCEN(15, 15),
YCEN(15, 15), ZCEN(15, 15), XCON(15, 15), YCON(15, 15),
ZCON(15, 15), AREA(15, 15), ARAT(15, 15), THETA(15, 15),
ALPHA(15, 15), CHORD(15, 15), XFOIL(16, 25, 2),
ZFOIL(16, 25, 2), XNUM(16, 2), XTAB1(25), ZTAB1(25),
XTAB2(25), ZTAB2(25)
```

IJ = Code for internal program control.

NPTS = Number of points on the successive wing percent chord lines that define the wing.

X, Y, Z = x, y, z coordinates of points on wing percent chord lines.

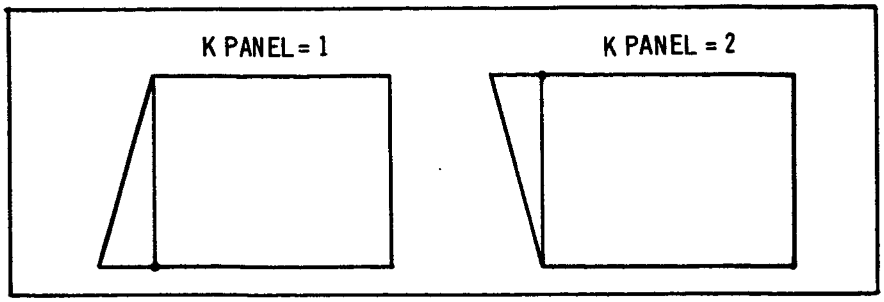
KPNT = Input constant

VALUE = Input array.

YCEPT = y-intercepts of wing cutting planes.

SLOPE = Slope (dx/dy) of outboard wing edge.

KPANEL = Code for wing panel type. If KPANEL (I, J) = 1, additional subpanel corner point is located on trailing edge. If KPANEL(I, J) = 2, point is located on leading edge. Note this code is determined <u>only</u> for panels on inboard column <u>if</u> body and wing intersect in nonstreamwise line, <u>and</u> for panels on outboard edge <u>if</u> outboard edge is a nonstreamwise line.




| | |
|---|---|
| XCOR, YCOR, ZCOR | = x, y, z coordinates of wing panel corner points. |
| XINT, YINT, ZINT | = x, y, z coordinates of additional wing panel point. |
| XCEN, YCEN | = x and y coordinates of wing panel centroid. |
| ZCEN: | = Wing panel thickness ordinate. |
| XCON, YCON | = x and y coordinates of wing panel control point. |
| ZCON | = Wing panel camber ordinate. |
| AREA | = Wing panel area. |
| ARAT | = Ratio of subpanel area to panel area. |
| THETA | = Wing panel thickness slope. |
| ALPHA | = Wing panel camber slope. |
| XFOIL, ZFOIL, XNUM | = Input arrays. |
| XTAB1, XTAB2, ZTAB1, ZTAB2 | = Temporary storage arrays used in subroutine slope. |

USAGE: COMMON (See subroutine OPCAMI for description of unlabeled COMMON)

CALL WING

SUBPROGRAMS
CALLED:

INPUTW
CRNRW
AREAP
CENTRD
CNTRLW
CHORDW
SLOPEW
OUTPTW

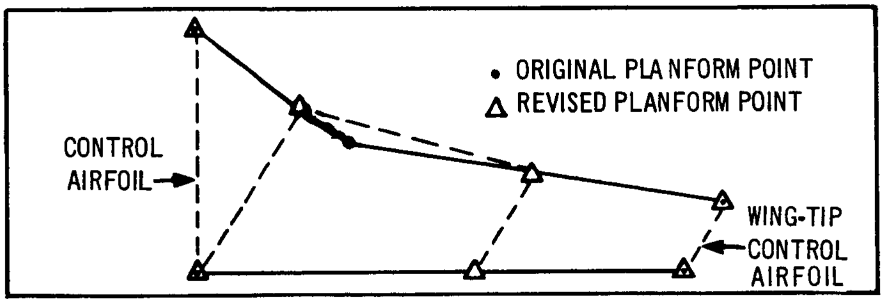
ERROR RETURNS: Error message indicates whether error occurred in calculation of wing panel geometry.

STORAGE: $220_{10} = 334_8$

**SUBJECT:** FORTRAN IV Subroutine WING1

**PURPOSE:** To read data cards that describe a wing by a planform and a set of control airfoils, and put this data in a standard form suitable for other subroutines.

**METHOD:** All of the data cards described in the attachment to subroutine WINGA are read. After the first three cards are read and checked, appropriate information is written on the output tape. Then storage is reserved in a buffer for variable length and scratch arrays (see Appendix A). Points describing the planform are read from cards 4 and 5. Subroutine WING1A is called to read cards 6, 7, and 8, scale the control airfoils to fit on the planform, and find the leading and trailing edge points corresponding to each airfoil. The original planform is assumed to be straight between given points; a new planform is constructed by joining the airfoil leading edge points and the trailing edge points. These planforms may differ as shown below.




After releasing scratch storage from the buffer, B, the percentages at which wing percent chord lines are to be constructed are read and checked. If an error is detected, a message is written.

**USAGE:**

DIMENSION B(1), DAT(2), KOD(2), TITLE(12), NU(3)

LOGICAL LOGUL(2)

CALL WING1 (B, LI, LO, DAT, LLETE, LOGUL, NAF, NPL, LAFN, LPCT, KOD, CHD, EPS, TITLE, NU)

Input: LI = Input tape number.

LO = Output tape number.

| | | |
|---|---|---|
| | DAT | = Date (alphameric). |
| | NU | = Error indicator array: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written. |
| Output: | B | = Storage buffer for variable-length arrays. |
| | LLETE | = Array containing the intersection points ($X_i$, $Y_i$) of the control airfoil with the wing leading and trailing edges is stored starting in B(LLETE). The point corresponding to the first airfoil, leading edge, is stored first, followed by the point for the trailing edge. The intersection points corresponding to the remaining airfoils are similarly stored. This array corresponds to the PT array in subroutine WING1A. |
| | LOGUL | = LOGUL(1) = .TRUE. if the upper wing surface is defined; LOGUL(2) = .TRUE. if the lower wing surface is defined. |
| | NAF | = Number of control airfoils. |
| | NPL | = Number of wing percent chord lines. |
| | LAFN | = An array, starting in B(LAFN) tells where, in B, the control airfoil points are stored and the number of elements (twice the number of points) in the airfoil upper or lower surface. This array is the same as array NAME in subroutine WING1A. The elements in this array are type INTERGER, although B is type REAL. The first four elements apply to the first airfoil, the second four elements to the second foil, etc. |

| | | |
|---|---|---|
| LPCT | = | Array of percentages (to specify set of wing percent chord lines) is stored starting in B(LPCT). |
| KOD | = | See ACODE and PCODE, card 3 (described in subroutine WINGA). |
| CHD | = | See CHD, card 2. |
| EPS | = | See EPS, card 3. |
| TITLE | = | See TITLE, card 1. |
| NU | = | Error indicator array. NU(1) is zero if no errors are detected. NU(3) may have been changed if an error was found (see Input, above). |

SUBPROGRAMS CALLED:

WING1A
MERR
IDLETE
IRSERV (See Appendix A)
IPACK

ERROR RETURNS: Function MERR is used to write an error message "ERROR i, CODE j IN SUBROUTINE WING1 DURING GEOMETRIC DEFINITION" if an error is detected.

| i | j | Explanation |
|---|---|---|
| 1 | 0 | WUL (see card 2) is not 0., 1., or 3. |
| 2 | 0 | PNLE, PNTE, AFN, or PLN (see card 2) is less than 2 or greater than 150. However, this is not considered a fatal error. |
| 3 | 0 | The storage buffer, B, is not large enough or a storage error has been detected. |
| 5 | 0 | |
| 8 | 0 | |
| 9 | 0 | |
| 7 | k | Error code k from subroutine WING1A. |
| 10 | 0 | A percentage (see card 9) is less than zero or greater than 100. |

RESTRICTIONS: None

STORAGE: $802_{10} = 1442_8$

SUBJECT: FORTRAN IV Subroutine WING1A

PURPOSE: To read data cards that define a set of control airfoils, to compute points on the airfoils (scaled to fit planform), and to find the intersections of each airfoil with the planform.

METHOD: The wing planform is given by a set of points $(X_i, Y_i)$ on the leading edge and another set on the trailing edge; adjacent points of each set are considered to be connected by straight lines. A card is read (card 6, see WINGA) that determines the placement of the airfoil on the planform and the number of points on the upper and lower surface of the airfoil. After these data are written, the intersections of the control airfoil and the planform leading and trailing edges are found. The airfoil chord length is the distance between these intersections.

A given code determines whether the upper wing, the lower wing, or both, are defined. Depending on this code, subroutine WING1B is called once or twice to read cards 7 and/or 8. These cards contain points $(U_i, V_i)$ which determine shape of the control airfoil. The points are scaled so that the magnitude of the difference of the U-coordinates of the first and last point is equal to the chord length (as computed from the planform). The scaled points have coordinates (U', Z) where U' is distance along the airfoil and Z is in the wing coordinate system. WING1B also writes airfoil data on the output tape. A control airfoil is considered to be a single point if only a single point is given or if the first and last given points have the same abscissa; in either case no scaling takes place, and the ordinate of the first point will be taken as Z. If the set of given control airfoil points contain $n > 2$ points and has a nonzero chord length, but the true chord length (from the planform) is zero, then the scaled airfoil will have n points with all coordinates (0, 0); this is an alternate way of specifying a point airfoil in the x,y plane.

USAGE:

```
DIMENSION  B(1), NAME(2, 2, 1), LPLF(2), NLTEP(3),
           PT(2, 2, 1), TITLE(12), DAT(2)

LOGICAL  LOGUL(2)

CALL WING1A (B, NAME, LPLF, NLTEP, LOGUL, PT,
             LI, LO, TITLE, DAT, MU)
```

Input: B = Storage buffer for variable length arrays.

LPLF = The points ($X_i$, $Y_i$) that define the leading edge of the wing are stored starting in B(LPLF(1)); similarly, the points on the trailing edge start in B(LPLF(2)).

NLTEP(1) = Number of points on the leading edge.

NLTEP(2) = Number of points on the trailing edge.

NLTEP(3) = Number of control airfoils = NAF.

LOGUL = LOGUL(1) = .TRUE. if the upper wing surface is to be defined; LOGUL(2) = .TRUE. if the lower wing surface is to be defined.

LI = Input tape number.

LO = Output tape number.

TITLE = Title (alphameric)

DAT = Date (alphameric)

Output: B = Storage buffer for variable length arrays.

NAME = Array that tells where in B the control airfoil points are stored, and the number of elements in each airfoil. The points ($U_i$, $Z_i$) for airfoil j are stored starting in B(NAME(1, i, j)), where i = 1 for upper surface of the airfoil, i = 2 for lower surface of the airfoil. The corresponding number of elements (twice the number of points) is stored in NAME(2, i, j).

PT = Array containing the intersections ($X_i$, $Y_i$) of the control airfoils with the leading and trailing edges of the wing. The X-coordinate of the intersection of control airfoil j with edge i is stored in PT(1, i, j), where i = 1 for leading edge, and i = 2 for trailing edge; similarly, the Y-coordinate is stored in PT(2, i, j).

MU = Error indicator, which is zero for success.

SUBPROGRAMS CALLED:

DISPTA
WING1B
SIN (Built-in function)
POLXN
COS (Built-in function)
IRSERV (See Appendix A)

ERROR RETURNS: See WINGA, card 6, for explanation of symbols.

| | |
|---|---|
| MU = 5 | AFK is not 1., 2., or 3. |
| MU = 6, 7 | YL (or YT) is outside the range of the leading edge (or trailing edge). If AFK = 1. or 2., MU = 6, if AFK = 3., MU = 7. |
| MU = 8 | The control airfoil (defined by AFK, BETA, YL, YT) does not intersect the leading edge (or trailing edge) in exactly one point. |
| MU = 9 | AFNU or AFNL is zero, indicating that this control airfoil has the same shape as the previous airfoil, but this is the first airfoil. |
| MU = 10 | The storage buffer, B, is not large enough to store the control airfoil points. |
| MU = 11 | This control airfoil has the same shape as the previous airfoil, but the previous one was a single-point airfoil. |

RESTRICTIONS: The given control airfoil points are not checked to see that they have reasonable values. The storage buffer, B, must have been initialized by subroutine INIBFR (Appendix A) or its equivalent.

STORAGE: $719_{10} = 1317_8$

**SUBJECT:** FORTRAN IV Subroutine WING1B

**PURPOSE:** To read and write data points describing a control airfoil and to scale both coordinates of each point so that the airfoil has a given chord length.

**METHOD:** After reading the control airfoil points (see NOTE, below), from an input tape, the point coordinates are written on an output tape. The chord length of a control airfoil is taken as the magnitude of the difference in X-coordinates of the first and last points. Then the airfoil is scaled by multiplying both coordinates of each point by the ratio of a given chord length to the computed chord length.

An option is provided so that instead of reading control airfoil points from cards, a given array of points is used; thereafter, the procedure is the same. This feature is useful when processing a series of airfoils which are similar to a preceding airfoil. Instead of writing out the (unscaled) points, the remark "THE GIVEN....AIRFOIL IS THE SAME AS PRECEDING ONE" is written.

An error code is returned if a control airfoil has less than two points or if the computed chord length is zero. Then the remark "THE AIRFOIL IS A POINT" is written. Since a single-point control airfoil may be legitimate, the calling program can take appropriate action.

**USAGE:** DIMENSION B(1), A(1)

CALL WING1B (LI, LO, UL, CHORD, NB, B, NA, A, NI

Input:

LI = Input tape number.

LO = Output tape number.

UL = Hollerith word "UPPER" or "LOWER."

CHORD = Chord length of the control airfoil.

NB = Number of elements (twice the number of points) in array B.

B = Array of points in a given control airfoil, stored ($x_1$, $z_1$, $x_2$, $z_2$, . . .). B is not used unless NA < 2.

NA = If NA < 2, this control airfoil is similar to the given airfoil, B, but has a chord length = CHORD; the airfoil points in B will

| | | |
|---|---|---|
| | | be used instead of being read from cards. If NA $\geq$ 2, NA is the number of elements (twice the number of points) in the control airfoil. NA elements will be read on tape LI. |
| Output: | NA | Twice the number of points in A. |
| | A | Array of NA/2 points in the control airfoil, scaled to chord length = CHORD, and stored $(x_1, z_1, x_2, z_2, \ldots)$. |
| | NU | Error indicator, which is zero for success. |

SUBPROGRAM CALLED: None

ERROR RETURNS: NU = -1 if the control airfoil has less than two points or if the first and last point have the same x-coordinate.
NU = 1 if NA is less than 2 (on input), and NB is less than 4.

RESTRICTIONS: A warning message is printed if the points in A are not increasing in x, but this condition is not considered to be an error.

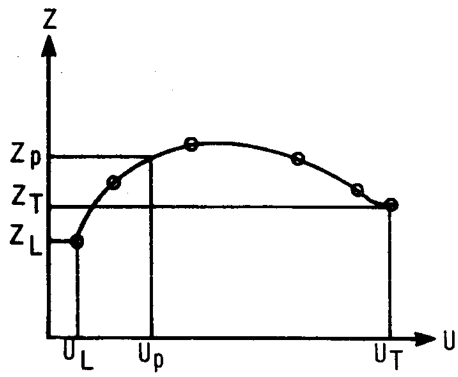
STORAGE: $332_{10} = 514_8$

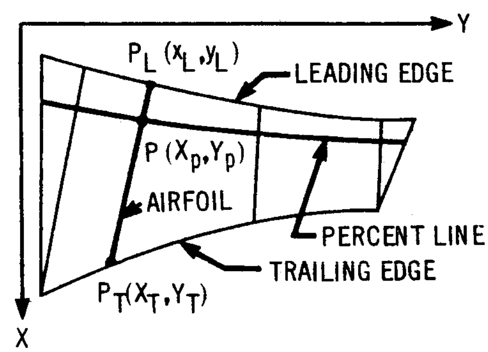
NOTE: See cards 7 or 8, Wing Input Data Format (attachment to subroutine WINGA).

**SUBJECT:** FORTRAN IV Subroutine WING2

**PURPOSE:** To construct three-dimensional points on wing percent chord lines (corresponding to given control airfoils), and, if required, to interpolate additional points.

**METHOD:** The input includes NAF control airfoils and NPL percent values. It is necessary to compute NAF points $(X_i, Y_i, Z_i)$ on each of the NPL percent chord lines. A point corresponding to a particular control airfoil and a percent chord line is found thus:




A set of points $(U_i, Z_i)$ which describe the control airfoil is illustrated in first figure above. The (X, Y) coordinates are given for points $P_L$ and $P_T$, shown in second figure. The control airfoil is to be placed on the planform so that $U_L$ corresponds to $P_L$, and $U_T$ to $P_T$; the U-coordinates of the airfoil may require scaling to do this, but the Z-coordinates are not to be scaled. Let p = percentage/100. Then

$$X_p = (1 - p) X_L + pX_T$$

$$Y_p = (1 - p) Y_L + pY_T$$

$$U_p = (1 - p) U_L + pU_T$$

The value of $Z_P$ is found by interpolation (using subroutine BITURP) from the control airfoil points ($U_i$, $Z_i$) at $U = U_P$, if a control airfoil is described by only one point, $X_P = X_L$, $Y_P = Y_L$, and $Z_P = Z_L$.

After the basic percent chord lines are computed, they are written on the output tape by subroutine WING2P. Subroutine RICHNA is then called to remove any near-coincident points in each percent chord line, to interpolate additional points (if CHD > 0.), and to place a "header" (table of contents) in front of the array. If both upper and lower wing surfaces are defined, the entire process is repeated for the lower surface. Then the control airfoil point arrays and the NAME array are deleted from buffer B, since they are no longer needed.

USAGE:

DIMENSION B(1), LOK(3,2), KOD(2), PLT(2,2,NAF), NAME(2,2,NAF), PCT(NPL), TITLE(12), DAYT(2), NU(3)

LOGICAL LGDEF(3,6), LOGUL(2)

CALL WING2 (B, LO, LGDEF, LOK, LOGUL, NAF, NPL, LAFN, KOD, PLT, NAME, PCT, CHD, EP, TITLE, DAYT, NU)

Input:

B = Storage buffer for variable length arrays.

LO = Output tape number.

LOGUL = LOGUL(1) = .TRUE. if upper wing is defined;

LOGUL(2) = .TRUE. if lower wing is defined.

NAF = Number of control airfoils.

NPL = Number of percent chord lines.

LAFN = The array NAME starts in B(LAFN); that is, the location in core of NAME (1,1,1) is the same as the location of B(LAFN).

KOD = Interpolation codes. KOD(1) is used by subroutine BITURP: KOD(2) is used by subroutine RICHNA. (See ACODE and PCODE, card 3, subroutine WINGA.)

PLT = Array of coordinates ($X_i$, $Y_i$) of intersection points of control airfoils with the leading and trailing edges of the

wing. The intersection point of the $k^{th}$ control airfoil with edge j is stored starting in PLT (1, j, k), where j = 1 for leading edge, = 2 for trailing edge. PLT corresponds to B(LLETE) in subroutine WING1.

NAME = An array that tells where in B the control airfoil points are stored, and the number of elements in each airfoil. The points ($U'_i$, $Z_i$) for control airfoil j are stored starting in B(NAME(1, i, j)), where i = 1 for upper surface of the airfoil, i = 2 for lower surface of the airfoil. The corresponding number of elements (twice the number of points) is stored in NAME(2, i, j). The NAME array is generated in subroutine WING1A.

PCT = Array of percentages associated with the wing percent chord lines.

CHD = Chord-arc tolerance. (See CHD, card 2, subroutine WINGA.)

EP = See EPS, card 3, subroutine WINGA.

TITLE = Title (alphameric).

DAYT = Date (alphameric).

NU = Error indicator array: NU(1) is not used, on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) < 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written.

Output: B = Storage buffer for variable-length arrays.

LGDEF = LGDEF(2,1) is set to .TRUE. if upper wing percent lines are successfully generated; LGDEF(2,2) is set to .TRUE. if lower wing percent chord lines are successfully generated.

LOK = Array which tells where in B the percent chord line arrays are located, and the length of each percent chord line array. Consider LOK(i, j). The upper wing surface corresponds to j = 1, the lower to j = 2. The percent chord line array, consisting of a header followed by point coordinates (see Appendix D), starts in B(LOK(1, j)). The number of percent chord lines in the array is stored in LOK(2, j). The total number of elements in the array is stored in LOK(3, j).

NU = Error indicator array. NU(1) is zero if no errors are detected. NU(3) may have been changed (see Input).

SUBPROGRAMS CALLED:

RICHNA
WING2P
BITURP
MERR
IDLETE, IRLEAS, IRSERV, IPACK (See Appendix A)

ERROR RETURNS: Function MERR is used to write a message "ERROR i, CODE j IN SUBROUTINE WING2 DURING GEOMETRIC DEFINITION" if an error is detected.

| i | j | Explanation |
|---|---|---|
| 1 | 0 | The storage buffer, B, is not large enough or a storage error has been detected. |
| 5 | 0 | |
| 6 | 0 | |
| 8 | 0 | |
| 2 | 0 | No points are given for a control airfoil (this error should be detected in a previous subroutine). |
| 3 | k | Error k from BITURP in interpolation from a given control airfoil to get a percent chord line point. The control airfoil has invalid points. |
| 7 | k | Error k from RICHNA. |

RESTRICTIONS: The storage buffer, B, must have been initialized by subroutine INIFBR (Appendix A) or its equivalent.

STORAGE: $550_{10} = 1046_8$

SUBJECT: FORTRAN IV Subroutine WING2P

PURPOSE: To write wing percent chord lines on an output tape.

METHOD: The basic wing percent chord lines, computed by subroutine WING2, consist of three-dimensional points on the surface of a wing. After title and headings are written on each page, the points are written, one point per line.

USAGE: DIMENSION W(3, NPT, NPCT), TITLE(12), DAYT(2), PCT(NPCT)

CALL WING2P (LO, W, NPCT, NPT, TITLE, UPLO, DAYT, PCT)

Input:

| | | |
|---|---|---|
| LO | = | Output tape number. |
| W | = | Array on NPCT percent chord lines. Each line contains NPT points ($x_i$, $y_i$, $z_i$). |
| NPCT | = | Number of percent chord lines. |
| NPT | = | Number of points in each percent chord line. |
| TITLE | = | Title (alphameric). |
| UPLO | = | 6H UPPER or 6H LOWER. |
| DAYT | = | Date (alphameric). |
| PCT | = | Array of percentages, corresponding to the percent chord lines in W. |

Output: Printout on tape LO.

SUBPROGRAM CALLED: None

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $205_{10} = 315_8$

| | |
|---|---|
| **SUBJECT:** | FORTRAN IV Subroutine WINGA |
| **PURPOSE:** | To read data cards defining a wing and construct percent chord lines on its surface. |
| **METHOD:** | Subroutine WING1 is called to read data cards (see attachment) that define the wing. Then subroutine WING2 constructs three-dimensional percent chord lines that lie on the wing surface. |
| **USAGE:** | DIMENSION B(1), DAT(2), LOK(3, 9), TITLE(2), NU(3) |
| | LOGICAL LGDEF(3, 6) |
| | CALL WINGA (B, LI, LO, DAT, LGDEF, LOK, LWPF, TITLE, NU) |

| | | |
|---|---|---|
| Input: | B | = Storage buffer for variable length arrays. |
| | LI | = Input tape number. |
| | LO | = Output tape number. |
| | DAT | = Date (alphameric). |
| | NU | = Error indicator array used by lower level routines: NU(1) is not used on input. NU(2) is an output tape number on which to write a message if an error is detected; no message is written if NU(2) $\leq$ 0. NU(3) is an error message limiter; if an error is detected, NU(3) = (NU(3) - 1). Then if NU(3) > 0 and NU(2) > 0, an error message is written. |
| Output: | B | = Storage buffer for variable length arrays. |
| | LGDEF(1, 1) | .TRUE. if upper wing was requested. |
| | (1, 2) | .TRUE. if lower wing was requested. |
| | (1, 4) | .TRUE. |
| | (2, 1) | .TRUE. if upper wing successfully defined. |
| | (2, 2) | .TRUE. if lower wing successfully defined. |

| | | |
|---|---|---|
| | (2, 4) | .TRUE. if WING1 was successful. |
| LOK | | = Array that tells where in B the wing percent chord line arrays and wing planform point array are located and the length of each array. Consider LOK (i, j). The upper wing percent chord line array corresponds to LOK(i, 1), the lower to LOK(i, 2); the planform point array corresponds to LOK(i, 7). An array starts in B(LOK(1, j)); the total number of elements in the array = LOK(3, j). LOK(2, 1) = number of upper percent chord lines, LOK(2, 2) = number of lower percent chord lines, and LOK(2, 7) = number of control airfoils = number of pairs of planform points (leading edge, trailing edge). |
| LWPF | | = The location of LWPF is the same as that of LOK(1, 7). (LWPF need not have been an argument.) |
| TITLE | | = Wing title (alphameric). |
| NU | | = Error indicator array. NU(1) is zero if no error is detected. NU(3) may have been changed (see INPUT, above). |

SUBPROGRAMS CALLED:

WING1
WING2
IPACK (See Appendix A )

ERROR RETURNS: NU(1) $\neq$ 0 if an error occurred in WING1 or WING2. LGDEF(2, 4) will be .FALSE. if the error was in WING1, .TRUE. if in WING2.

RESTRICTIONS: The storage buffer, B, must have been initialized by subroutine INIBFR (Appendix A) or its equivalent. LGDEF(i, j), i = 1, 3; j = 1, 2, 4 must have been set to .FALSE.

STORAGE: $153_{10} = 231_8$

ATTACHMENT: Wing Input Data Format

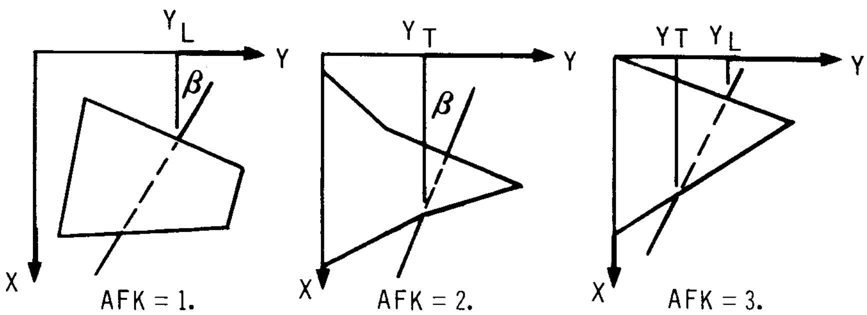
| | Column | Code | Explanation |
|---|---|---|---|
| Card 1 | 1-72 | TITLE | Any desired title. |
| Card 2 | 1-10 | PNLE | Number of corner or break points defining the planform leading edge. |
| | 11-20 | PNTE | Number of corner or break points defining the planform trailing edge. |
| | 21-30 | AFN | Number of planform control airfoils. AFN $\geq$ 2. including the wing tip airfoil. |
| | 31-40 | PLN | Number of constant percent chord lines. PLN $\geq$ 1. |
| | 41-50 | WUL | = 1. only the upper wing surface is defined.<br><br>= 2. only the lower wing surface is defined.<br><br>= 0. both upper and lower wing surfaces are defined. |
| | 51-60 | CHD | Dimensional tolerance for the density of points interpolated (between control airfoils) on percent chord lines. If CHD $\leq$ 0., or if AFN $<$ 4., no additional points are generated. Do not specify CHD $>$ 0. unless every basic percent chord line has at least four distinct points. |
| Card 3 | 1-10 | PCODE | Code that controls the type of three-dimensional interpolation to be used for generating additional points on percent chord lines. PCODE is not effectively used on any basic percent chord line of less than four distinct points or if CHD $\leq$ 0.. For a more detailed explanation of PCODE, see KD in subroutine RICH3A. |

| Column | Code | Explanation |
|---|---|---|
| | | = 0., enriching (interpolation of additional points) is done on points projected into a plane which "best" fits the basic percent chord line points. Enriched percent lines can appear curved between control airfoils in the planview. |
| | | = 1., enriching is done on points projected into a plane normal to the X-Y plane through the percent chord line ends. Resulting percent lines appear as straight lines between control airfoils in the planview. |
| | | = 2., enriching is done as for PCODE = 1. but resultant percent lines can appear curved between control airfoils in the planview. |
| | | NOTE: If PCODE = 1. or 2., and the percent chord line lies roughly in a horizontal plane (e.g., leading or trailing edge), enriching may not add points to the basic percent line. |
| 11-20 | ACODE | = 1., linear interpolation is used between points on control airfoils to find points on the basic percent chord lines. |
| | | = 2., biquadratic interpolation is used. |
| 21-30 | EPS | A tolerance used to eliminate coincident percent chord line points. The basic percent chord lines consist only of intersections of percent chord lines with control airfoils. If two of these points are closer together than EPS, one is eliminated. If EPS $\leq$ 0., a value of 0.01 is used. |

<u>Cards 4</u>

| Column | Code | Explanation |
|---|---|---|
| 1-10 | $X_1$ | Array of points defining the planform leading edge, arranged in order from inboard to outboard. There must be PNLE point pairs, three coordinate |
| 11-20 | $Y_1$ | |
| 21-30 | $X_2$ | |

| | Column | Code | Explanation |
|---|---|---|---|
| | 31-40 | $Y_2$ | pairs per card. Note that the leading edge may be modified by the program, depending on the placement of control airfoils on the planform. |
| | 41-50 | $X_3$ | |
| | 51-60 | $Y_3$ | |
| Cards 5 | 1-10 | $X_1$ | Similar to cards 4, but for the trailing edge. |
| | 11-20 | $Y_1$ | |
| | 21-30 | $X_2$ | |
| | 31-40 | $Y_2$ | |
| | 41-50 | $X_3$ | |
| | 51-60 | $Y_3$ | |
| Cards 6 | | | These cards, one for each control airfoil, give the airfoil location on the planform and the number of points on the upper and lower parts of the control airfoil. Each card 6 is followed by one or more cards 7 and 8 (unless the previous airfoil is to be repeated). |
| | 1-10 | AFK | Code to indicate how the control airfoil is oriented on the planform. See sketches below. |




| | | | |
|---|---|---|---|
| | | | Two of the three quantities $\beta$ , $y_L$, $y_T$ must be given to specify the control airfoil location. AFK indicates which two quantities are to be used. |
| | 11-20 | BETA | The angle $\beta$, positive if as shown on the sketches. BETA is ignored if AFK = 3.. |
| | 21-30 | $Y_L$ | $Y_L$, Y-coordinate at which the control airfoil intersects the leading edge. $Y_L$ is ignored if AFK = 2.. |

| | Column | Code | Explanation |
|---|---|---|---|
| | 31-40 | $Y_T$ | $Y_T$, Y-coordinate at which the control airfoil intersects the trailing edge. $Y_T$ is ignored if AFK = 1.. |
| | 41-50 | AFNU | Number of points on the control airfoil upper surface. The coordinates of these points (cards 7) follow this card. However, if AFNU = 0., then the previous airfoil upper surface points are used and a card 7 must not follow. If WUL = 2. (upper wing not defined), then AFNU is ignored, and a card 7 must not follow. |
| | 51-60 | AFNL | Number of points on the control airfoil lower surface. The coordinates of these points (cards 8) follow cards 7 (or card 6 if no card 7 is used). However, if AFNL = 0., then the previous airfoil lower surface points are used and a card 8 must not follow. If WUL = 1. (lower wing not defined), then AFNL is ignored, and a card 8 must not follow. Note that the leading and trailing edge points must be included, even if they repeat those of the airfoil upper surface. |
| Cards 7 | 1-10 | $U_1$ | Array of points ($U_i$, $V_i$) on the control airfoil upper surface, in increasing order from leading edge to trailing edge. The points will later be scaled to fit the planform. This card must not follow a card 6 if WUL = 2. or if AFNU = 0.. |
| | 11-20 | $V_1$ | |
| | 21-30 | $U_2$ | |
| | 31-40 | $V_2$ | |
| | 41-50 | $U_3$ | |
| | 51-60 | $V_3$ | |
| Cards 8 | 1-10 | $U_1$ | Similar to card 7, but apply to the control airfoil lower surface points. Cards 8 follow cards 7 (or a card 6 if a card 7 is not used). A card 8 must not be used if WUL = 1. or if AFNL = 0.. |
| | 11-20 | $V_1$ | |
| | 21-30 | $U_2$ | |
| | 31-40 | $V_2$ | |
| | 41-50 | $U_3$ | |
| | 51-60 | $V_3$ | |

| | Column | Code | Explanation |
|---|---|---|---|
| Cards 9 | 1-10 | $P_1$ | Array of constant percent chord values corresponding to the spanwise panel edges. For example, if percent chord lines 0, 20, 50, 100 are desired, then $P_1 = 0.$, $P_2 = 20.$, $P_3 = 50.$, and $P_4 = 100.$. |
| | 11-20 | $P_2$ | |
| | . | | |
| | . | | |
| | 51-60 | $P_6$ | |

SUBJECT: FORTRAN IV Subroutine WLDM

PURPOSE: To compute the coefficients of lift, drag, and pitching moment of the wing and the spanwise distribution of the coefficients of lift and drag.

METHOD: The force normal to a panel is given as the product of dynamic pressure, surface pressure coefficient, and panel are:

$$F_i = q \cdot C_{p_i} \cdot A_i$$

Resolving into the components normal and parallel to the free-stream direction yields

$$L_i = - F_i \cdot \cos \theta_i$$

$$D_i = F_i \cdot n_i$$

where $n_i$ is the component of velocity normal to the x-axis and $\theta_i$ is the angle between the plane of the panel and a plane parallel to the x-y plane.

The moment of force with respect to a point (x, 0, z) gives the pitching moment.

$$M_i = - L_i (\bar{x}_i - x) + D_i (\bar{z}_i - z)$$

where $\bar{x}$ and $\bar{z}$ are the x and z coordinates of the panel centroids.

The coefficients of lift, drag, and moment for the entire wing are given by

$$C_L = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} L_i$$

$$C_D = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} D_i$$

$$C_M = \frac{1}{q \cdot S_W} \sum_{i=1}^{NM} M_i$$

where $S_W$ is the wing reference area, and NM is the number of wing panels.

The spanwise distribution of the coefficients of lift and drag is given by

$$C_{L_j} = \frac{1}{q \cdot S_W} \sum_{i=1}^{NROW} L_i$$

$$C_{D_j} = \frac{1}{q \cdot S_W} \sum_{i=1}^{NROW} D_i \quad , j = 1, \ldots, NCOL$$

where NROW is the number of panels in a chordwise column, and NCOL is the number of chordwise columns.

USAGE:

CALL WLDM (NM, NROW, XP, ZP, RFAREA, AREA, XBAR, ZBAR, ALPHAM, THETAM, CPM, SCL, SCD, CL, CD, CM)

DIMENSION AREA(NM), XBAR(NM), ZBAR(NM), ALPHAM(NM), THETAM(NM), CPM(NM), SCL(NCOL), SCD(NCOL)

Input:

- NM = Number of wing panels.
- NROW = Number of panels in a steamwise column.
- XP = x-coordinate of the point about which the pitching moments are to be calculated.
- ZP = z-coordinate of the point about which the pitching moments are to be calculated.
- RFAREA = Wing reference area.
- AREA = Array of wing panel areas.
- XBAR = Array of wing panel centroid x-coordinates.
- ZBAR = Array of wing panel centroid z-coordinates.
- ALPHAM = Array of velocity components normal to the x-axis.
- THETAM = Array of angles between the plane of the panel and a plane parallel to the x-y plane.

Output:

- SCL = Array of spanwise lift coefficients.
- SCD = Array of spanwise drag coefficients.
- CL = Coefficient of lift.

| | | |
|---|---|---|
| CD | = | Coefficient of drag. |
| CM | = | Coefficient of pitching moment. |

SUBPROGRAM CALLED: COS

ERROR RETURNS: None

RESTRICTIONS: None

STORAGE: $187_{10} = 273_8$

The Boeing Company
Commercial Airplane Division
Renton, Washington
August 1967

# 5. APPENDIXES

## 5.1 Appendix A — Dynamic Storage Allocation Package

SUBJECT: Dynamic Storage Allocation in FORTRAN

PURPOSE: To provide a means for allocating, using, and releasing blocks of storage on a dynamic basis during execution of FORTRAN programs.

METHOD: The buffer format is illustrated in figure 1, page 421. Each buffer is divided into a (variable) number of arrays, some of which may be full while others are available to be filled. Initially the buffer contains only one array; this is empty. The first word of each buffer is a "header" which contains the total length of the buffer.

Each array is preceded by a "secret" header of two codes, illustrated in figure 1, which contains the following information:

1) Whether the array is full or empty

2) The length of the array

3) The relative address of the first word of the buffer

4) The absolute address of the array name (if the array is full).

In using the buffer, a programmer can ignore the headers, because the subroutines he uses will accommodate this bookkeeping. These routines will also indicate situations such as an overflowing buffer or reference to an array in the buffer that is not a legitimate array.

The subroutines involved in the scheme may be divided into three classes:

1) Basic subprograms

2) Primitive subprograms

3) User's subprograms

The first two of these classes are an integral part of the automatic storage scheme and are discussed in more detail below. Their structure is illustrated in figure 2, page 422. The user's subprograms are provided by the authors of the computing system and will call the basic subprograms into execution. The user should never have to call a primitive subprogram directly. When setting up an array in the buffer, either by use of the function IADARY or by reserving space with the function IRSERV, the user will obtain a "name" for that array stored as a FORTRAN integer in a cell that may be called NAME. This name is now uniquely

associated with the array. If the buffer is called BUFFER, the first word of the array will always be BUFFER(NAME) and the $i^{th}$ word will be BUFFER(NAME + I - 1), regardless of any shuffling that may take place automatically in the buffer. However, because data shuffling may occur, the array must always be referred to by the same NAME, which will be kept updated automatically. Obviously, each array can have only one name.

USAGE:

**Basic Subprograms:**

The following seven subprograms form the basic set which can be called directly by the user. With the exception of INIBFR and CLEAR they are coded in FORTRAN.

1) Subroutine CLEAR (BFRA, BFRB, ...)

   To reinitialize and make available for reuse any number of buffers BFRA, BFRB, ..., etc.

2) Subroutine INIBFR (BFRA, NA, BFRB, NB, ...)

   To initialize any number of buffers, BFRA, etc., of length NA, etc., words. This routine must be called before a buffer can be used.

3) Function IADARY (ARRAY, N, BFR, NAME)

   To place an array, ARRAY, of length N words, into the buffer, BFR, and to assign to it the name, NAME. The function value will be the same as NAME unless some error or misuse is detected, in which case it will be negative. To circumvent the oddities of the FORTRAN subscripting, procedure IADARY should be used thus:

   NAME = IADARY (ARRAY, N, BFR, NAME)

4) Function IDLETE (BFR(NAME))

   To delete (i.e., make available for reuse) an array in the buffer, BFR, whose name is NAME. The function value will be zero unless some error is detected, when delection will not occur.

5) Function IPACK (BFR)

   The pack a buffer, BFR, by moving all full arrays to the beginning of the buffer and condensing all the empty arrays into one at the end. The function value will be equal to the length of the final empty array unless an error is detected, when it will be negative and packing will not occur.

6) Function IRLEAS (BFR(NAME), N)

To release all cells beyond cell N in an array in the buffer, BFR, whose name is NAME; i.e., BFR(NAME) - BFR(NAME + N-1) are retained but BFR(NAME + N) - the end of the array are made available for reuse. The function value will be zero unless some error is detected, when release will not occur.

7) Function IRSERV (N, BFR, NAME)

To reserve an array of N words in the buffer, BFR, and to assign to it the name, NAME. The comments under IADARY concerning function value apply also to IRSERV.

Primitive Subprograms: For completeness, the eight primitive subprograms of the scheme are briefly described here. They are all coded in assembly language.

1) Function ARFRE (NAME)

To set up the first code word of an empty array. FORTRAN IV version has deck name ARFREE.

2) Function ARFUL (NAME)

To set up the first code word of a full array.

3) Function ARSEC (N, NAME)

To set up the second code word of an array. FORTRAN IV version has deck name ARSECR.

4) Function FOOL(I)

To store a fixed point variable in a floating point location. The FORTRAN IV version has deck name UFOOL and two entry points, FOOL and IFOOL.

5) Function IBFR (A)

To locate the first word of a buffer relative to the first word of an array.

6) Function ICODE (A)

To determine whether an array is full or empty.

7) Function LENG (A)

To determine the length of an array

8) Subroutine UPDATE (A)

To update the contents of an array name following data shuffling in the buffer.

EXAMPLE: An example is given in figure 3, page 423, of a program which reads in an ($\ell$ x m) and an (m x n) matrix, multiplies them together and prints out the ($\ell$ x n) resultant. The input arrays are then deleted from the buffer.

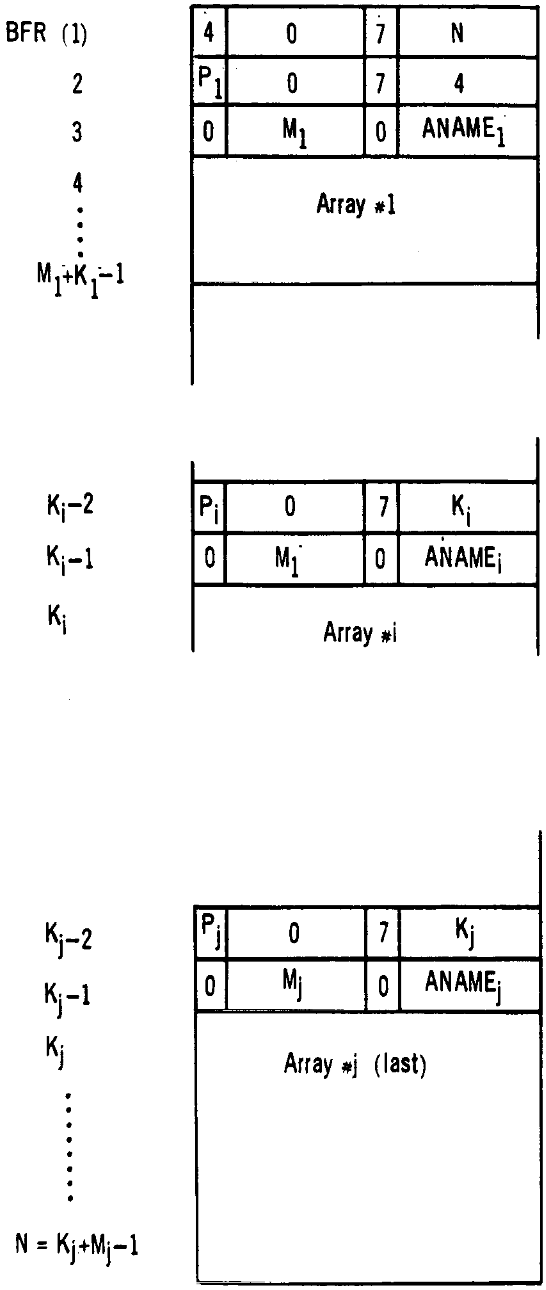
FLOWCHARTS: Flowcharts of all the primitive and basic subprograms of the dynamic storage package are included. These flowcharts are for the FORTRAN II versions. The FORTRAN IV subprograms differ slightly because of the forward storage of arrays and full word instead of decrement format of fixed-point variables.

SUBPROGRAM CALLED: None

EQUIPMENT: IBM 7094 II

STORAGE:

| Subprogram | FORTRAN II Octal | FORTRAN II Decimal | FORTRAN IV Octal | FORTRAN IV Decimal |
|---|---|---|---|---|
| ARFREE | 10 | 8 | 7 | 7 |
| ARFUL | 10 | 8 | 7 | 7 |
| ARSECR | 10 | 8 | 11 | 9 |
| CLEAR | 25 | 21 | 22 | 18 |
| IADARY | 273 | 187 | 255 | 173 |
| IBFR | 24 | 20 | 23 | 19 |
| ICODE | 23 | 19 | 22 | 18 |
| IDLETE | 65 | 53 | 62 | 50 |
| INIBFR | 33 | 27 | 33 | 27 |
| IPACK | 316 | 206 | 266 | 182 |
| IRLEAS | 173 | 123 | 165 | 117 |
| IRSERV | 42 | 34 | 45 | 37 |
| LENG | 4 | 4 | 5 | 5 |
| UFOOL | 2 | 2 | 2 | 2 |
| UPDATE | 14 | 12 | 13 | 11 |
| TOTALS | 1334 | 732 | 1252 | 682 |




NOTES

For a full array:

$P_i = 7$

$ANAME_i$ is the absolute address of the array name, $NAME_i$. The word $NAME_i$ contains $K_i$ as a decrement integer.

For an empty array:

$P_i = 0$

$ANAME_i$ is irrelevant.

In general:

| | |
|---|---|
| $M_i$ | is the length of the ith array. |
| $K_i$ | is the first word of the array relative to the beginning of the buffer. |
| N | is the total length of the buffer. |

The first code word of the buffer and of the arrays are illegal both as FORTRAN integers and as floating point numbers.

FIGURE 1 BUFFER FORMAT

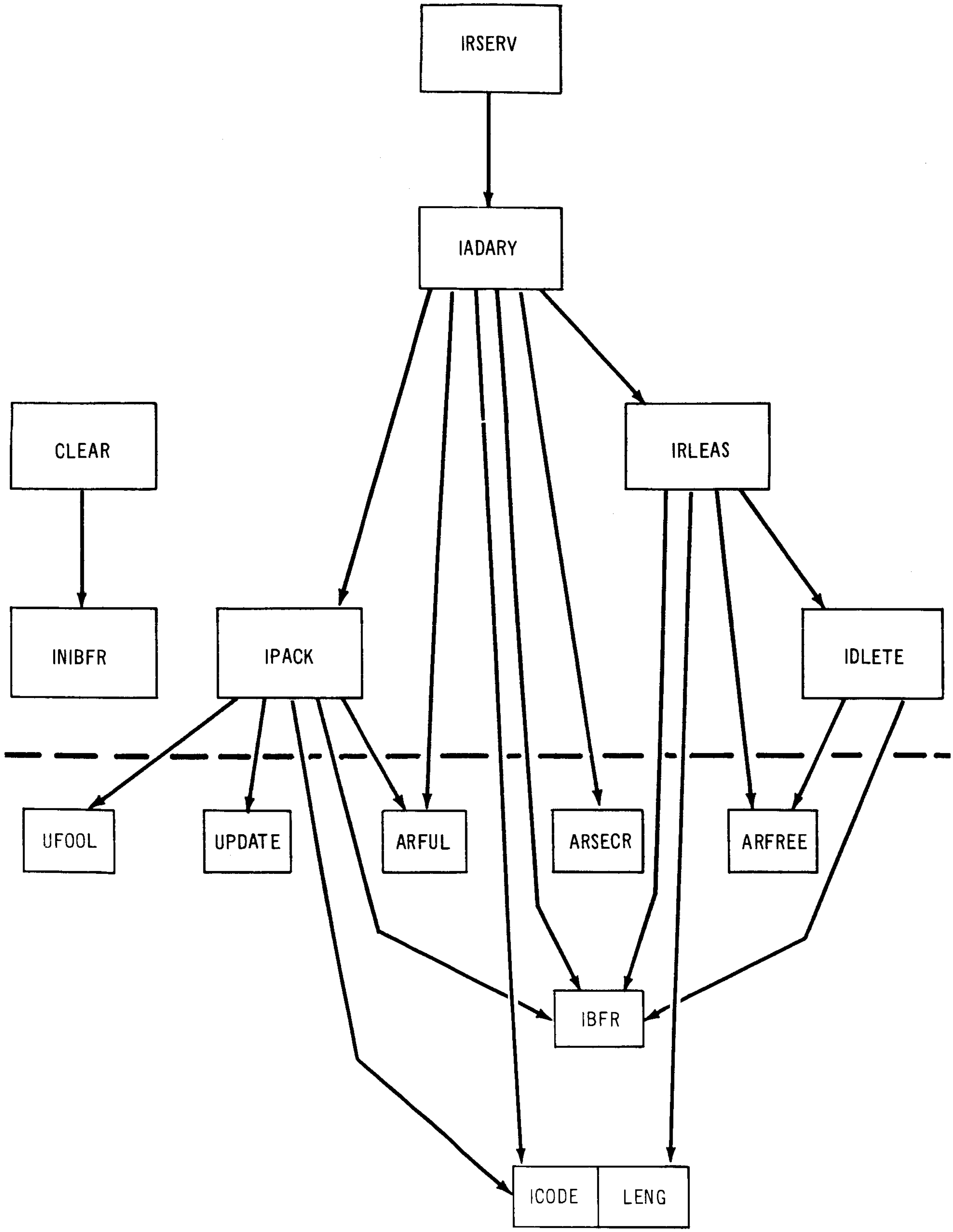
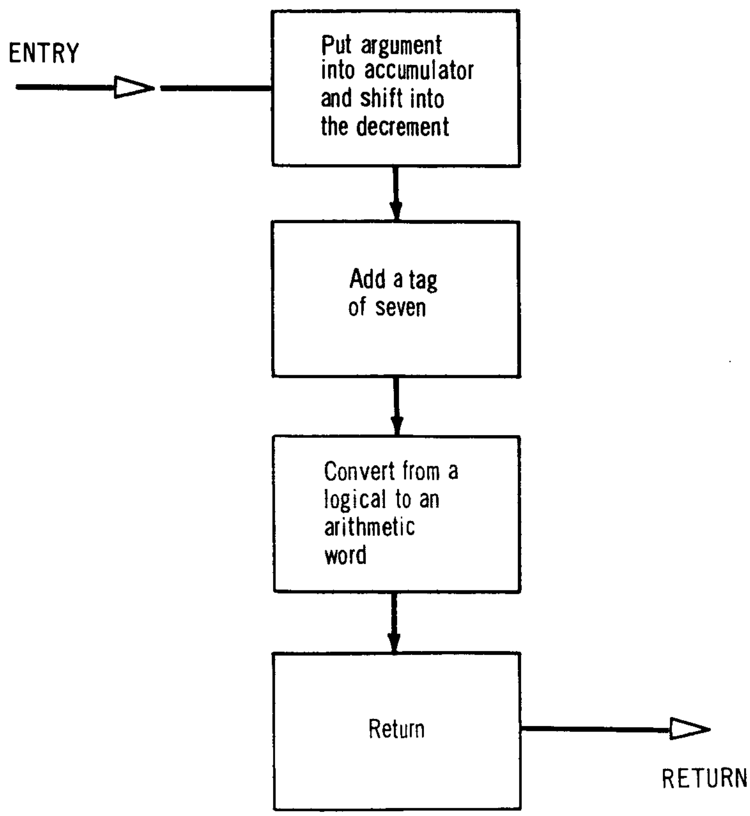



FIGURE 2 STRUCTURE OF THE DYNAMIC STORAGE SCHEME

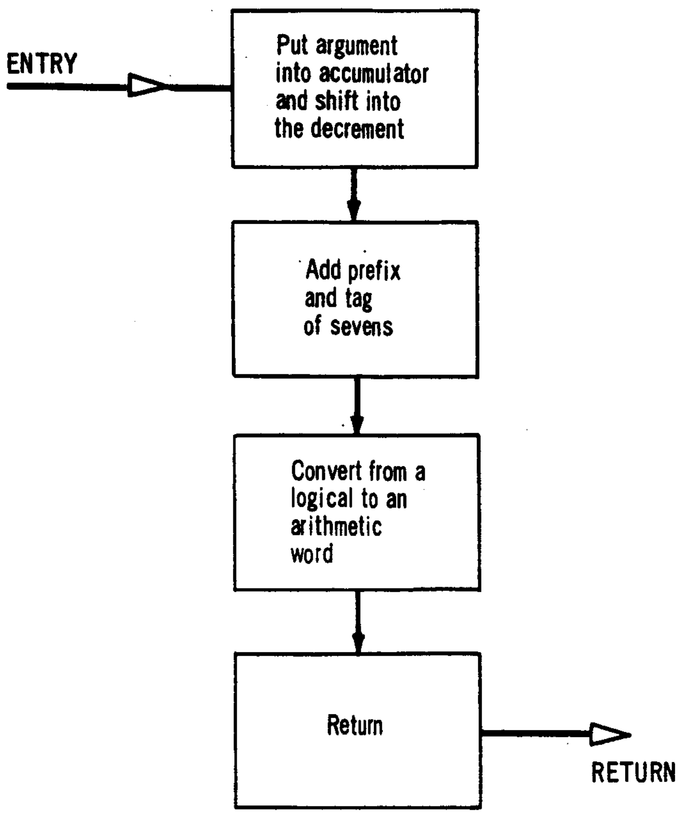
```
      DIMENSION BFR(1000)
C
      CALL INIBFR (BFR,1000)
C
      READ INL
      READ INPUT TAPE 5,1,L,M,N
    1 FORMAT (6G)
C
      NAM1=IRSERV(L*M,BFR,NAM1)
      NAM2=IRSERV(M*N,BFR,NAM2)
      IJ=IRSERV(L*N,BFR,NAM3)-1
C
      K=NAM1+L*M-1
      READ INPUT TAPE 5,1,(BFR(I),I=NAM1,K)
      K=NAM2+M*N-1
      READ INPUT TAPE 5,1,(BFR(1),I=NAM2,K)
C
      DO 20 J=1,N
      DO 20 I=1,L
      IK=NAM1+I-1
      KJ=NAM2+(J-1)*M
C
      DO 10 K=1,M
      B=B+BFR(IK)*BFR(KJ)
      IK=IK+L
   10 KJ=KJ+1
C
      IJ=IJ+1
   20 BFR(IJ)=B
C
      CALL IDLETE(BFR(NAM1))
      CALL IDLETE(BFR(NAM2))
C
      WRITE OUTPUT TAPE 6,2,(BFR(I),I=NAM3,IJ)
    2 FORMAT (G20.6)
      STOP
      END(1,1,0,0,0,0,0,1,0,1,0,0,0,0,0)
```

FIGURE 3 EXAMPLE FOR DYNAMIC STORAGE

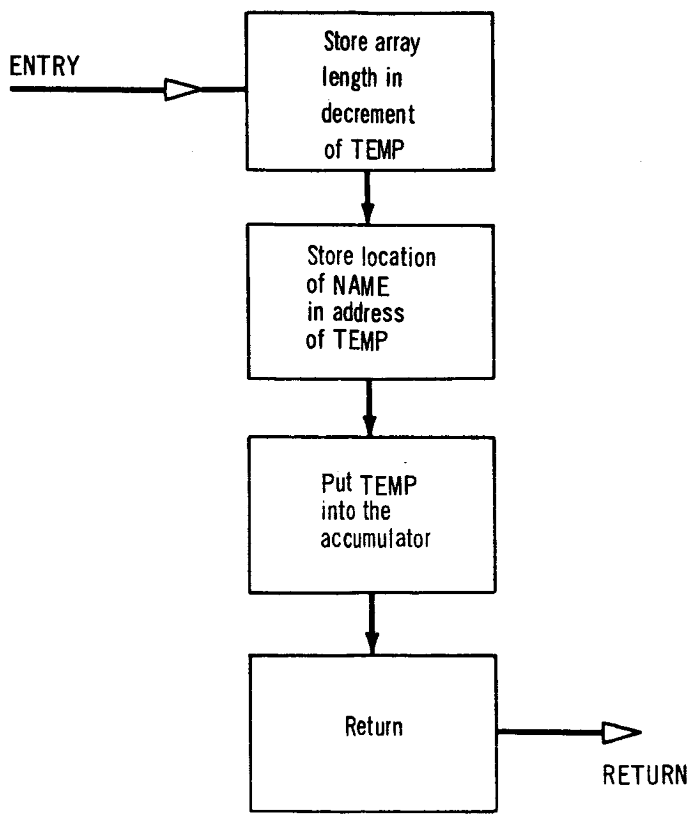
ARFREE


ENTRY
Put argument into accumulator and shift into the decrement
Add a tag of seven
Convert from a logical to an arithmetic word
Return
RETURN

ARFUL

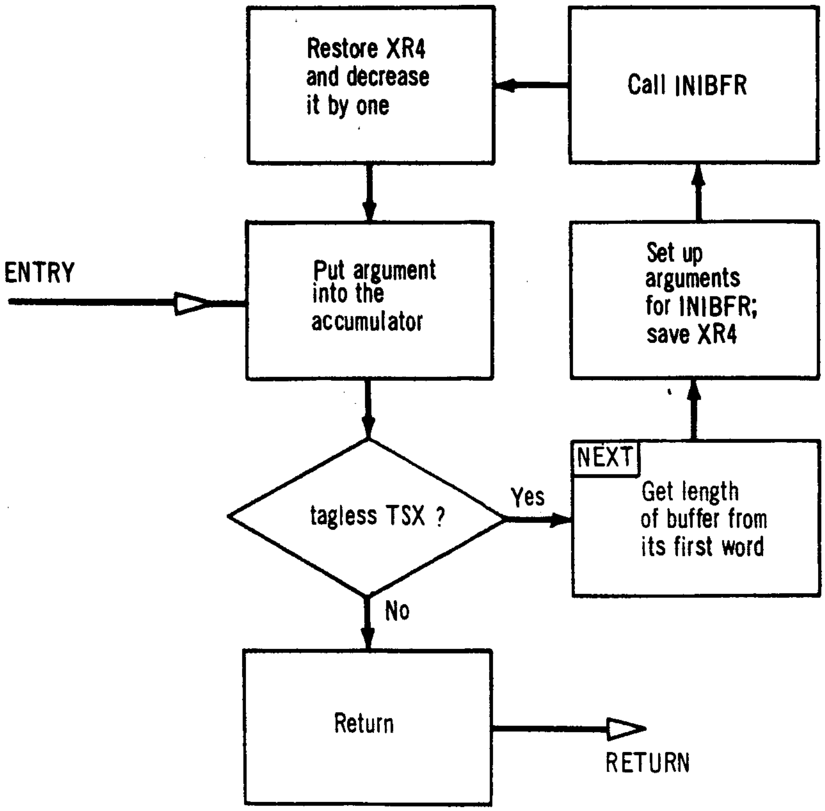

ENTRY
Put argument into accumulator and shift into the decrement
Add prefix and tag of sevens
Convert from a logical to an arithmetic word
Return
RETURN

ARSECR

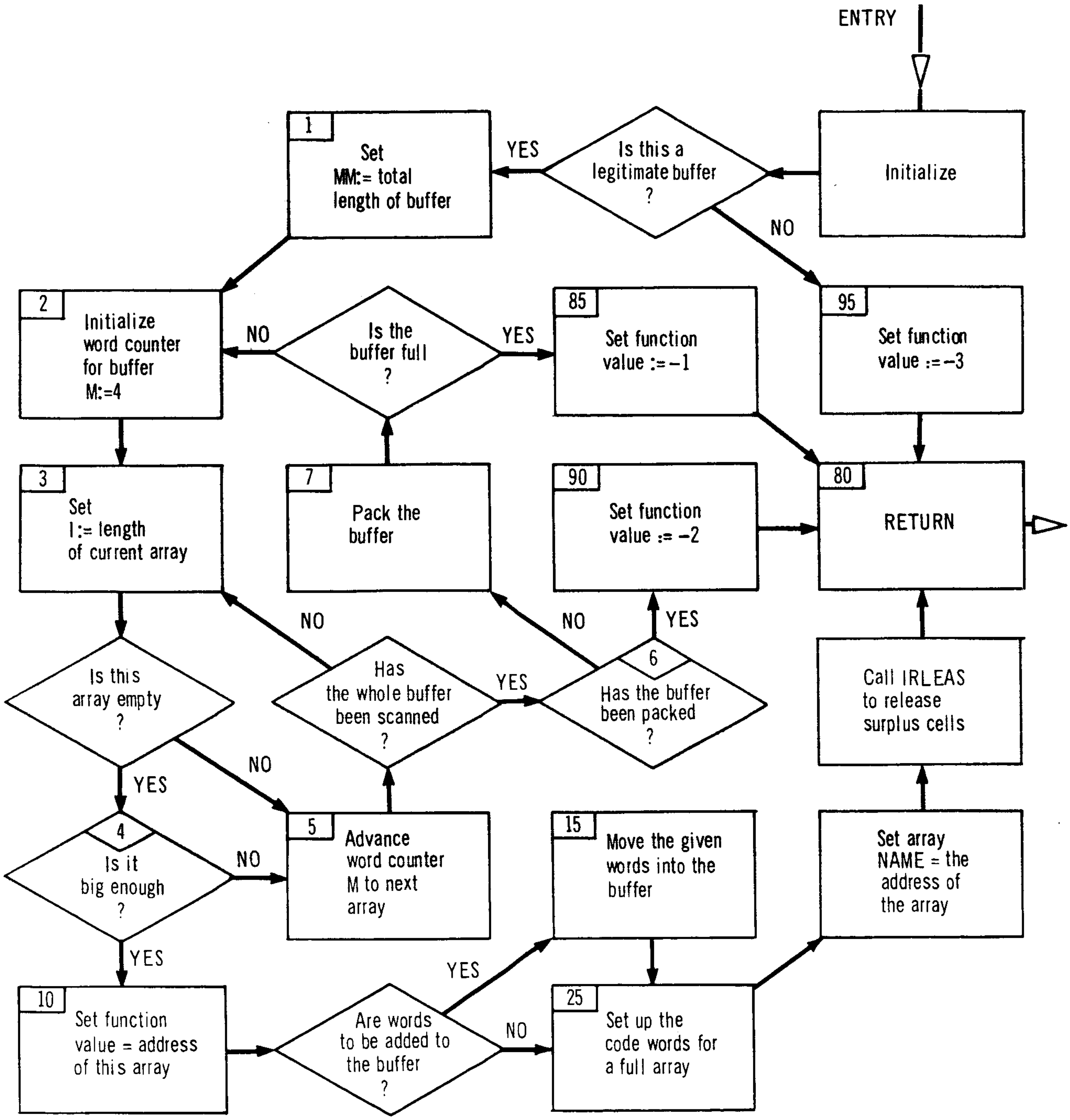


ENTRY
Store array length in decrement of TEMP
Store location of NAME in address of TEMP
Put TEMP into the accumulator
Return
RETURN

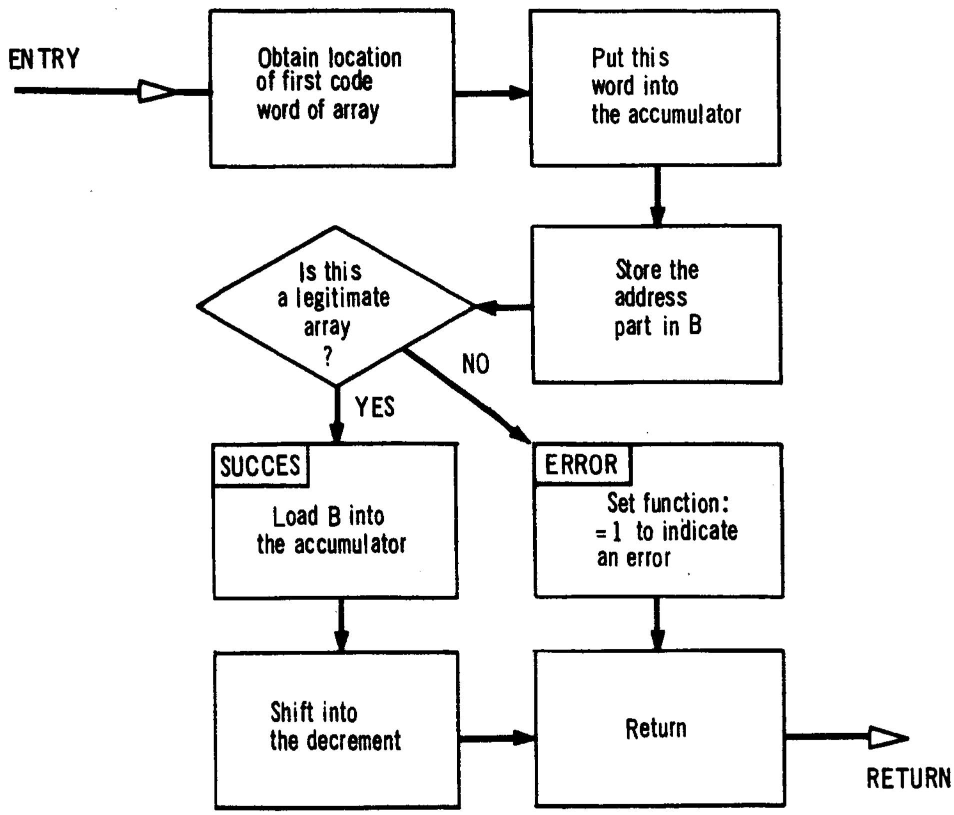
CLEAR


Restore XR4 and decrease it by one
Call INIBFR
ENTRY
Put argument into the accumulator
Set up arguments for INIBFR; save XR4
tagless TSX ?
Yes
NEXT
Get length of buffer from its first word
No
Return
RETURN

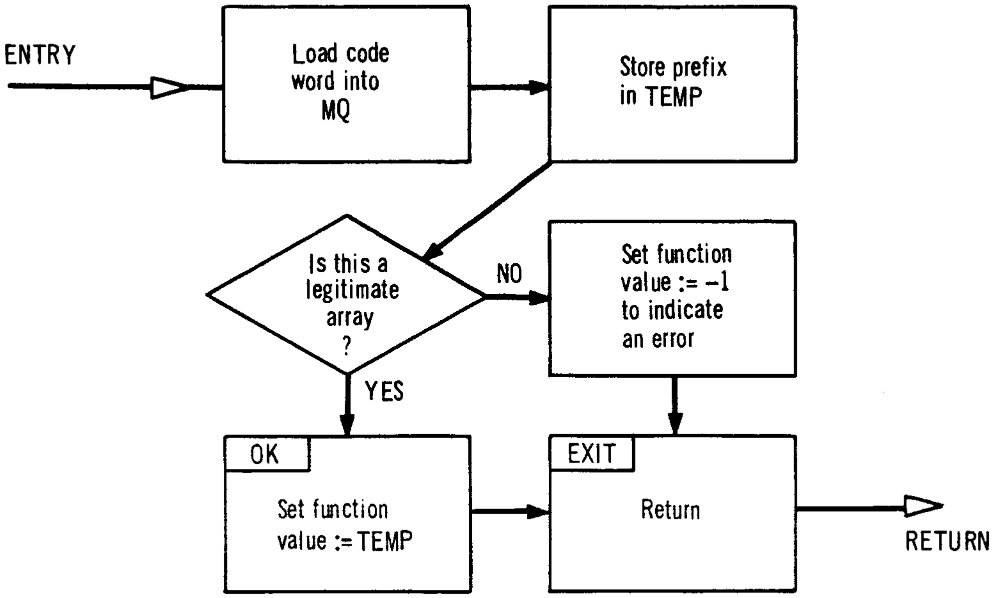
IADARY


ENTRY
Initialize
Is this a legitimate buffer ?
YES
NO
1
Set MM:= total length of buffer
95
Set function value :=−3
2
Initialize word counter for buffer M:=4
NO
Is the buffer full ?
YES
85
Set function value :=−1
80
RETURN
3
Set I:= length of current array
7
Pack the buffer
90
Set function value := −2
NO
NO
YES
Is this array empty ?
Has the whole buffer been scanned ?
YES
6
Has the buffer been packed ?
Call IRLEAS to release surplus cells
NO
YES
4
Is it big enough ?
NO
5
Advance word counter M to next array
15
Move the given words into the buffer
Set array NAME = the address of the array
YES
YES
10
Set function value = address of this array
Are words to be added to the buffer ?
NO
25
Set up the code words for a full array

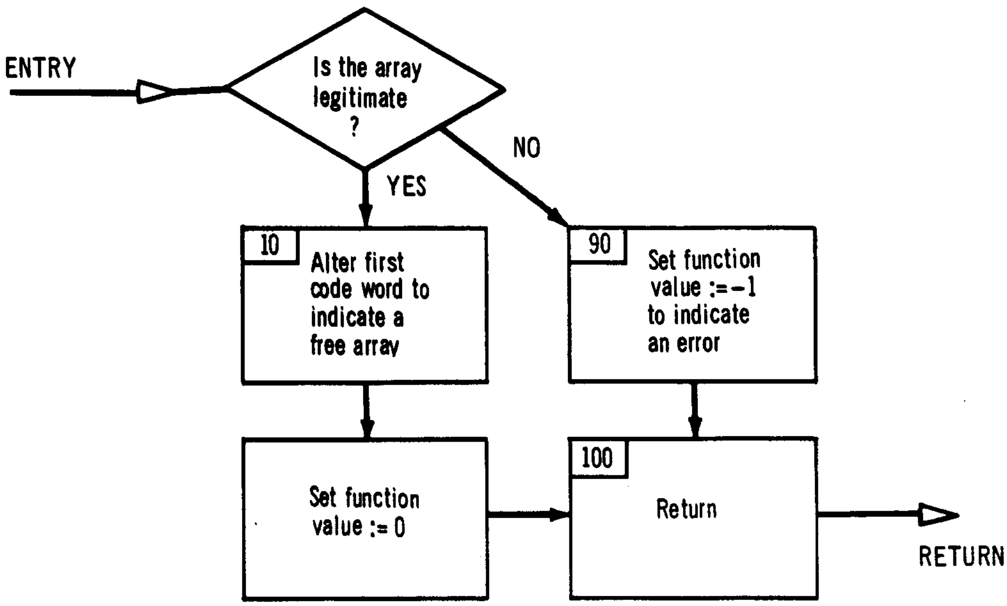
IBFR


ENTRY
Obtain location of first code word of array
Put this word into the accumulator
Store the address part in B
Is this a legitimate array ?
NO
YES
SUCCES
Load B into the accumulator
ERROR
Set function: =1 to indicate an error
Shift into the decrement
Return
RETURN

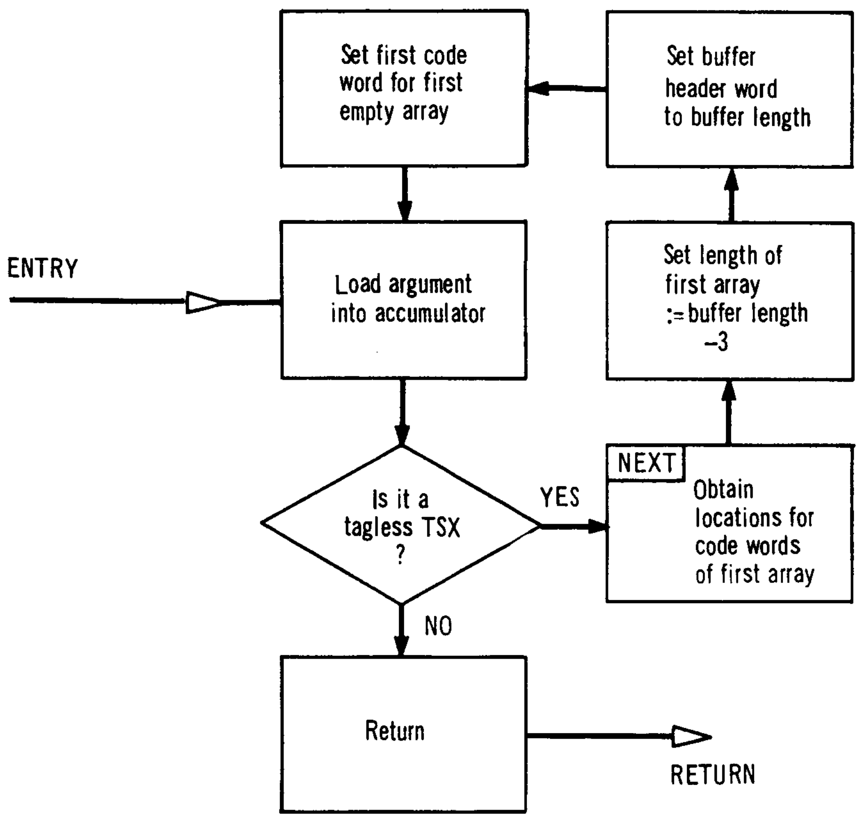
ICODE


ENTRY
Load code word into MQ
Store prefix in TEMP
Is this a legitimate array ?
NO
Set function value := −1 to indicate an error
YES
OK
Set function value := TEMP
EXIT
Return
RETURN

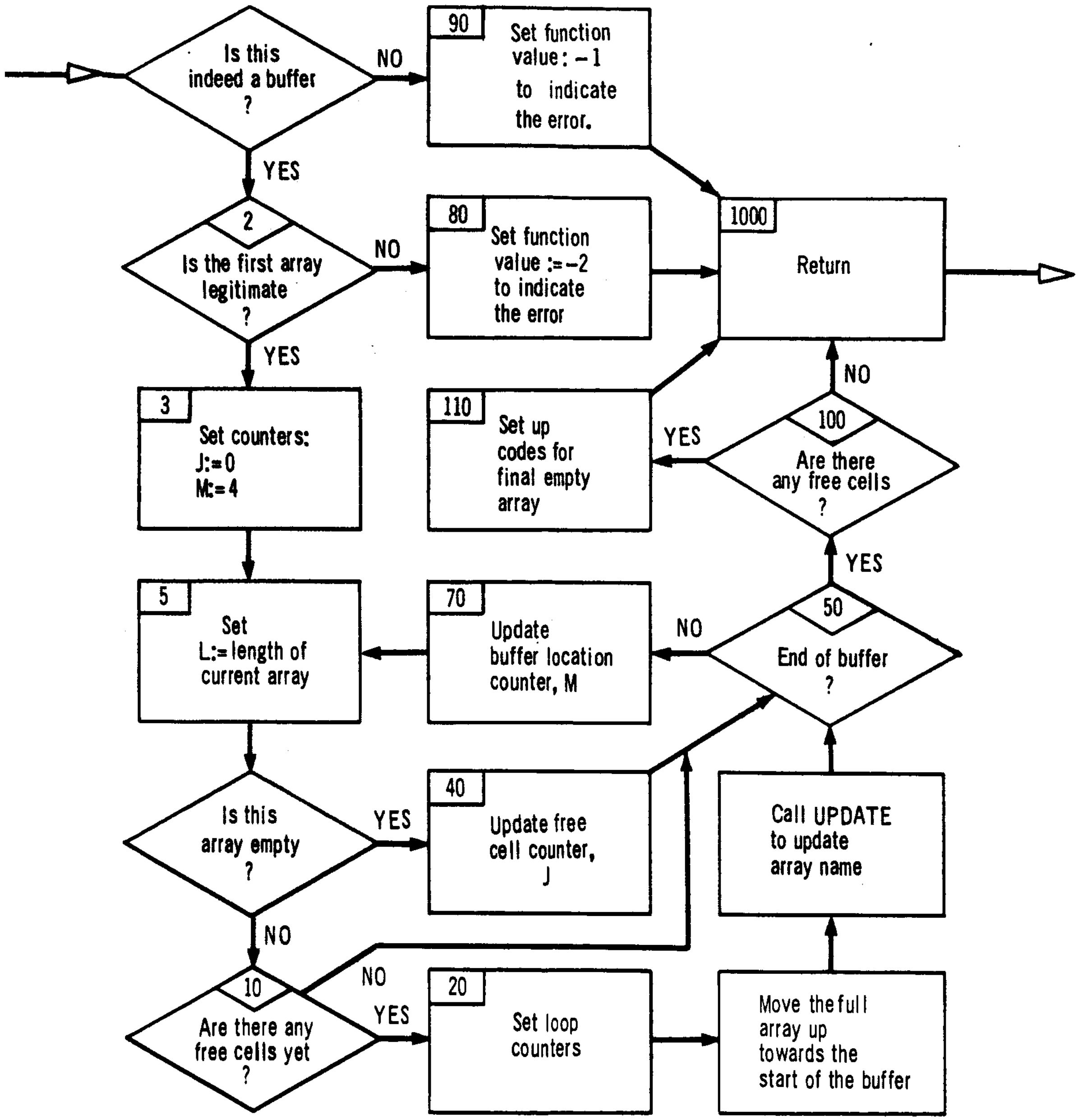
IDLETE


ENTRY
Is the array legitimate ?
YES
NO
10
Alter first code word to indicate a free array
90
Set function value :=−1 to indicate an error
Set function value := 0
100
Return
RETURN

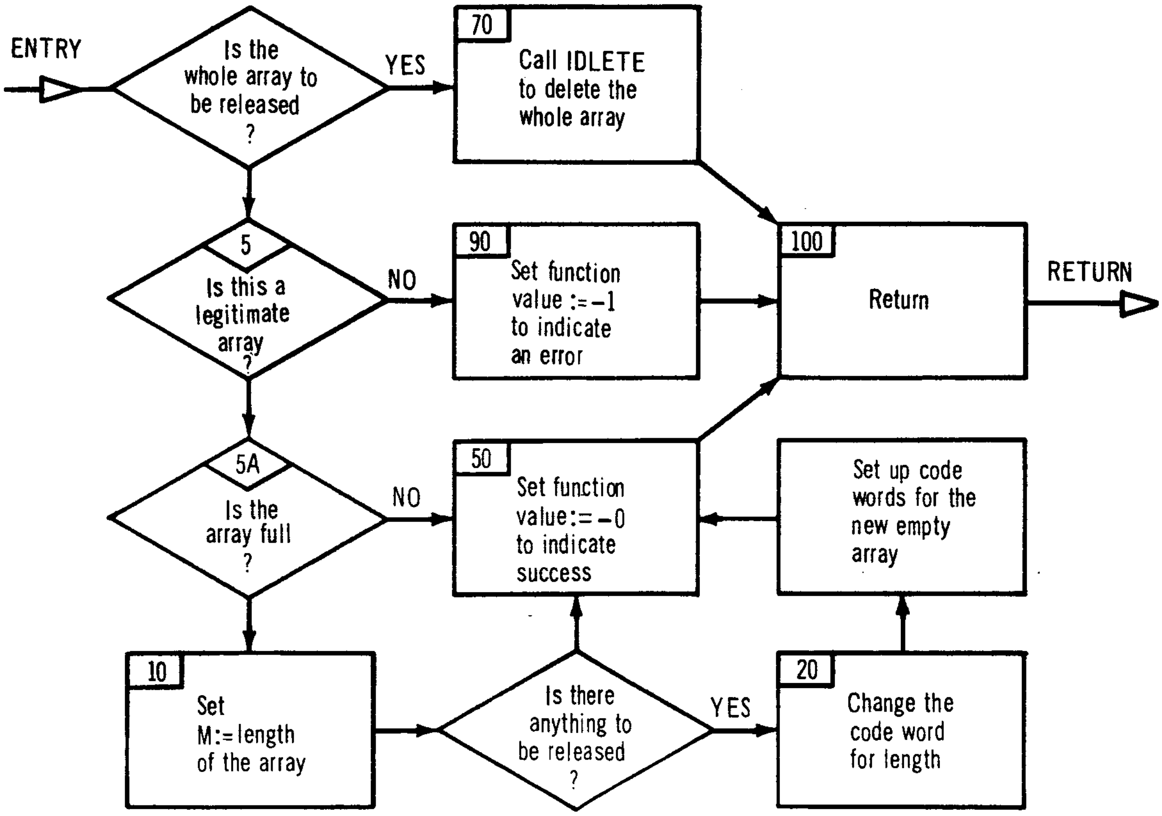
INIBFR


Set first code word for first empty array
Set buffer header word to buffer length
ENTRY
Load argument into accumulator
Set length of first array :=buffer length -3
NEXT
Obtain locations for code words of first array
Is it a tagless TSX ?
YES
NO
Return
RETURN

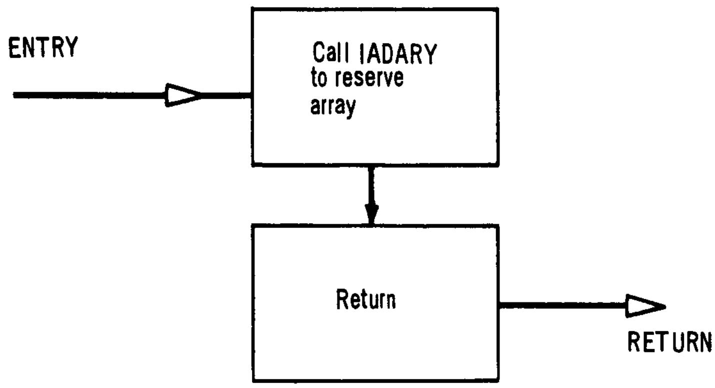
IPACK


Is this indeed a buffer ?
NO
YES
90
Set function value: −1 to indicate the error.
1000
Return
2
Is the first array legitimate ?
NO
YES
80
Set function value :=−2 to indicate the error
3
Set counters: J:=0 M:=4
110
Set up codes for final empty array
YES
NO
100
Are there any free cells ?
YES
5
Set L:=length of current array
70
Update buffer location counter, M
NO
50
End of buffer ?
Is this array empty ?
YES
40
Update free cell counter, J
Call UPDATE to update array name
NO
10
Are there any free cells yet ?
NO
YES
20
Set loop counters
Move the full array up towards the start of the buffer

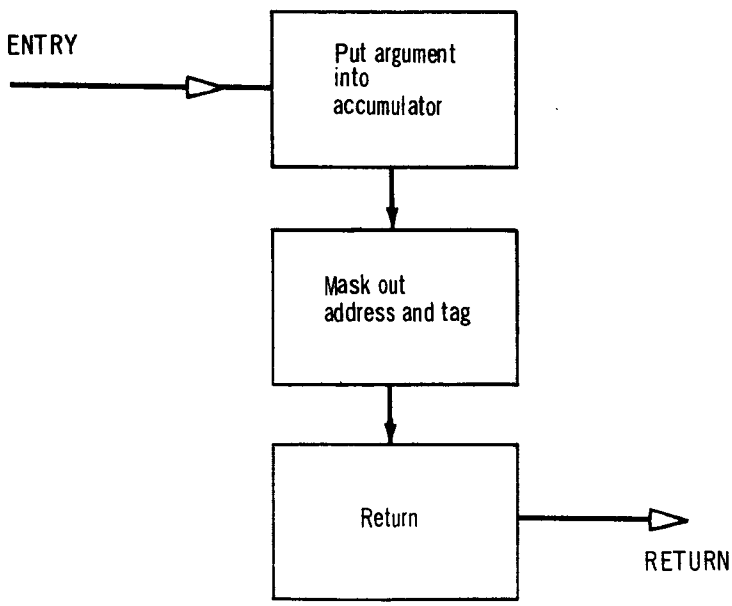
IRLEAS


ENTRY
Is the whole array to be released ?
YES
70
Call IDLETE to delete the whole array
5
Is this a legitimate array ?
NO
90
Set function value :=−1 to indicate an error
100
Return
RETURN
5A
Is the array full ?
NO
50
Set function value:=−0 to indicate success
Set up code words for the new empty array
10
Set M:=length of the array
Is there anything to be released ?
YES
20
Change the code word for length

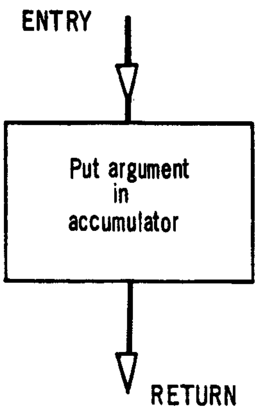
IRSERV



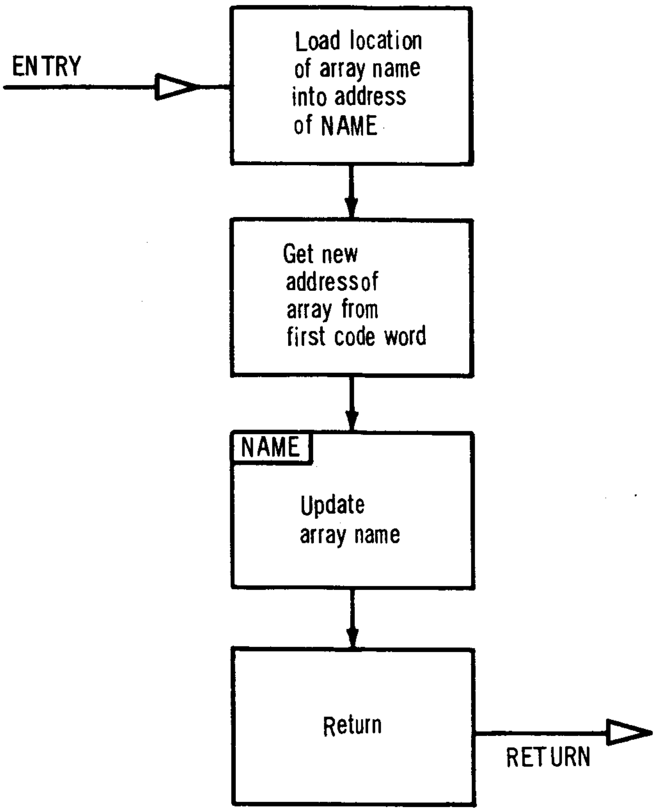
LENG


ENTRY
Put argument
into
accumulator
Mask out
address and tag
Return
RETURN

UFOOL



UPDATE


ENTRY
Load location of array name into address of NAME
Get new addressof array from first code word
NAME
Update array name
Return
RETURN

## 5.2 Appendix B — Matrix Inversion Package

SUBJECT: FORTRAN IV Subroutine SINVRT and MAP Subroutine INV4S

PURPOSE: To invert an n-by-n matrix and compute its determinant.

METHOD:

1. Before the inversion process starts, the matrix is scaled so that the element of largest magnitude is greater than or equal to 1/4 and less than 1/2. This is accomplished through fixed-point addition or subtraction in the characteristic, which avoids the introduction of round-off during scaling.

2. A theorem of matrix algebra states that for any non-singular matrix A, there exist elementary transformations R and S such that:

$$RAS = I \tag{1}$$

   where I is the identity matrix. Because R and S are elementary transformations, they are necessarily non-singular which permits taking the inverse of both sides of equation (1).

$$S^{-1} A^{-1} R^{-1} = I$$

   Then solving for $A^{-1}$ gives,

$$A^{-1} = SIR = SR$$

   The transformations S and R are determined by reducing A to the identity matrix. In INV4S R consists of row interchanges and arithmetic operations on rows, while S consists solely of column interchanges.

3. The inversion is performed in single precision, and it leaves a single-precision inverse and determinant. Normal 7094 single-precision floating-point restrictions apply.

4. The inversion of a sparse matrix is speeded up by a checking feature that can sense a zero reduction factor and bypass a row reduction.

5. At the start of each reduction stage, rows and columns are interchanged to select the element of largest magnitude from the remaining submatrix and place it in the pivotal position.

USAGE:

CALL SINVRT (A, K, N, IRR1, IRR2, SCALE, DET, NDETXP)

| | | |
|---|---|---|
| A | = | Matrix to be inverted. |
| K | = | Row DIMENSION statement entry for A. |
| N | = | Number of rows (columns) in matrix (decrement integer). |
| IRR1 | = | Error code from INV4S (decrement integer). |
| IRR2 | = | Error code from INV4S (decrement integer). |
| SCALE | = | Error cell from INV4S. |
| DET | = | Mantissa part of determinant (single precision): $(1. < \|DET\| < 10.)$ |
| NDETXP | = | Exponent part of determinant (decrement integer): $-(2^{17} - 1) < NDETXP < (2^{17} - 1)$ |
| determinant | = | $DET \times 10^{(NDETXP)}$ |

CALL INV4S (A, N, IRR1, IRR2, SCALE, DET, NDETXP)

| | | |
|---|---|---|
| A | = | Location of (1, 1) matrix element. |
| N | = | Number of rows (columns) in matrix (address integer). |
| IRR1 | = | Error code 1 (address integer). |
| IRR2 | = | Error code 2 (address integer). |
| SCALE | = | Error cell. |
| DET | = | Mantissa part of determinant (single precision): $(1. < \|DET\| < 10.)$ |
| NDETXP | = | Exponent part of determinant: $-(2^{35} - 1) < NDETXP < (2^{35} - 1)$ |
| determinant | = | $DET \times 10^{(NDETXP)}$ |

SUBPROGRAMS CALLED:

A. SINVRT calls INV4S

B. INV4S None

ERROR RETURNS: Error codes for both subroutines:

| | | |
|---|---|---|
| IRR1 | = 0 | inversion successful |
| | = 1 | overflow occurred |
| | = 2 | matrix is singular |
| | = 3 | scaled inverse cannot be rescaled without causing overflow |
| | = 4 | rows and columns cannot be rearranged |
| | = 10 + I | error code 3 occurred subsequent to error code 1 |
| | = 20 + I | error code 4 occurred subsequent to error code 1 |

IRR2 = Rank if IRR1 = 2
= Reduction stage during which overflow occurred if IRR1 = 1
= 0 for all other values of IRR1

SCALE = Scale factor if IRR1 = 3
= 0 for all other values of IRR1

If IRR1 = 1 or 2, then the matrix will be rearranged and rescaled as though the inversion were successful, and the determinant will be set to zero.

RESTRICTIONS:

A. These must apply to the two routines:

1. They must be used on the IBM 7094 only.
2. The matrix is always destroyed during the inversion process, since the inverse is stored over the matrix.
3. Because of the internal scaling process, some matrix elements will be set to zero if the ratio of the largest to smallest magnitudes exceeds $10^{38}$.
4. If underflow occurs, a normal zero is stored for the answer. Overflow always results in an error return.
5. The machine is left in single tag mode and floating point trap mode upon return to the calling program.

B. These apply to SINVRT only:

1. The inversion is performed by INV4S.
2. The matrix to be inverted must be stored in normal FORTRAN IV order and in single precision.

3. During the inversion process a total of $n^2 + 2n + 3$ locations must be available starting at the matrix (1, 1) element and ending at a lower core location.

C. This applies to INV4S only:

The total storage required is $n^2 + 2n + 3$.

STORAGE:

A. SINVRT $315_{10} = 473_8$

B. INV4S $489_{10} = 751_8$

TIMING:

For INV4S and SINVRT, the time in seconds to invert a matrix of order n is given by:

$$\text{Time} = 4.0 \times 10^{-5} \times n^3 \text{ seconds}$$

CHECKOUT:

All of the routines were tested on various-order Hilbert matrices and their inverses. Below is a table of the results.

| Hilbert Matrix Order | Minimum Number of Significant Figures in Inverse (INV4S/SINVRT) |
|---|---|
| 2 | 8 |
| 3 | 7 |
| 4 | 5 |
| 5 | 4 |
| 6 | 3 |

Also tested was the PEI matrix (Comm. A.C.M. V.5, 1962, page 508) with various diagonal terms and various orders. All input was single precision.

| Order of PEI Matrix | Diagonal Term | Minimum Number of Significant Figures in Inverse (SINVRT) |
|---|---|---|
| 40 | 1.001 | 5 |
| 60 | 1.001 | 6 |
| 65 | 1.00001 | 3 |
| 100 | 1.001 | 5 |
| 100 | 1.0001 | 4 |
| 100 | 1.00001 | 3 |
| 100 | 1.000001 | 2 |
| 100 | 1.0000001 | 2 |
| 100 | 1.5 | 7 |
| 100 | 1000. | 7 |

| Order of PEI Matrix | Diagonal Term | Minimum Number of Significant Figures in Inverse (SINVRT) |
|---|---|---|
| 130 | 1000. | 6 |
| 130 | 1000. | 6 |

RECOMMENDED USAGE:

Below is a portion of a FORTRAN IV program which uses SINVRT acceptably.

```
      DIMENSION  A(60, 60), B(123), D(3723)
      EQUIVALENCE (A, C), (B, C(3601))
      .
      .
      .
      READ (5, 8000) N, ((A(I, J), J=1, N), I=1, N)
      CALL SINVRT (A, 60, N, IRR1, IRR2, SCALE, DET,
                   NDETXP)
      IF (IRR1) 6000,20,6000
   20
      .
      .
 6000 WRITE (6, 9000)IRR1, IRR2, SCALE

      CALL EXIT
 8000 FORMAT  (I10/(7F10.0))
 9000 FORMAT  (1H1,5X, 18H ERROR CODE 1 = I5, 14H
             ERROR CODE 2=I5, 14H SCALE FACTOR =
             E16.8)
```

The array C has $N^2 + 2N + 3 = 3723$ cells to ensure sufficient core to invert A. An extra 123 locations are then available in B for use at other times.

## 5.3 Appendix C — Differential Equation Solver Package

**SUBJECT:** FORTRAN IV Subroutines DED, DEDIS, DEI, DES

**PURPOSE:** To solve a set of N first-order ordinary differential equations by the Adams-Moulton predictor-corrector method.

**METHOD:** The differential equations must be in the form:

$$y'_1 = f_1 (t, y_1, y_2, \ldots, y_n)$$

$$y'_2 = f_2 (t, y_1, y_2, \ldots, y_n)$$

----------------------------

$$y'_1 = f_i (t, y_1, y_2, \ldots, y_i, \ldots, y_n)$$

$$y'_n = f_n (t, y_1, y_2, \ldots, y_n)$$

t is the independent variable and $y_1, y_2, \ldots y_n$ are the dependent variables that must be determined for a given range of t .

A description of this method follows:

1. The initial conditions for $t, y_1, y_2, \ldots, y_n$ must be given as well as an initial dependent variable increment h . Backward integration may be performed by using a negative h .

2. Each differential equation requires four points before the predictor-corrector equations may be applied. The first three integrations are performed by the Runge-Kutta method, and the initial conditions give the required starting points for the predictor-corrector formulas. Each time the step size is altered, it is necessary to obtain new starting points using Runge-Kutta integration.

3. Once four points are determined, the integration is performed from point n to n + 1 along the curve using equation 3.a to predict and 3.b to correct. Let h be the independent variable increment:

a. $y^{(p)}_{i,n+1} = y_{i,n} + \frac{h}{24} (55f_{i,n} - 59f_{i,n-1}$

$+ 37f_{i,n-2} - 9f_{i,n-3})$

b. $y^{(p)}_{i,n+1} = y_{i,n} + \frac{h}{24} (9f_{i,n+1} + 19f_{i,n}$

$- 5f_{i,n-1} + f_{i,n-2})$

(p) and (c) represent predicted and corrected values respectively. $y^{(p)}_{i,n+1}$ and $y^{(c)}_{i,n+1}$ are tested. If the test is satisfied so that it is not required to change h, the derivatives $y'_{i,n+1}$ are calculated from the differential equations.

USAGE:

More then one system of differential equations may be solved in a given FORTRAN program. Let N be the maximum number of differential equations required to solve in a FORTRAN program. The following must be considered in using the routines.

1. A FORTRAN floating array of 12A + 9 storage cells must be reserved in a dimension statement. This block shall be referred to as T.

2. Dealing with a particular system of differential equations, the number of equations must be stored in two FORTRAN fixed variable locations. These locations will be referred to as NUM and NUM1.

3. An equivalence statement giving T(1) and NUM1 the same working location must be written.

4. There are certain parameters which are input to the differential equation routine. The following parameters are used:

   $\beta$ — The fractional part to decrease the interval h if it needs to be reduced. Normally, set $= 0.5$ and it must be in the range $0 < \beta < 1$.

   A — A number used in determining the relative error between the predicted and corrected values. Normally, A = 1.

   $h_{min}$ — The minimum increment of the independent variable. This value must not be negative.

$h_{max}$ The maximum allowable increment of the independent variable. This value must not be negative.

M Ratio between upper bound and lower bound relative error. Normally, give it the value 1,000.

E Allowable relative error between the predicted and corrected value. A representative value is $10^{-6}$.

5. The following FORTRAN locations must be initialized with the indicated quantities prior to entering the differential equation solving routines.

| FORTRAN location | |
|---|---|
| NUM, NUM1 | Number of differential equations that must be solved. |
| T(12*NUM) | Initial value of $1^{st}$ dependent variable. |
| T(12*NUM-1) | Initial value of $2^{nd}$ variable. |
| . | |
| . | |
| . | |
| T(12*NUM-NUM+1) | Initial value of last dependent variable. |
| T(12*NUM+1) | Initial increment of independent variable. |
| T(12*NUM+2) | Initial value of independent variable. |
| T(12*NUM+4) | $\beta$ as defined in 4. |
| T(12*NUM+5) | $h_{min}$ |
| T(12*NUM+6) | $h_{max}$ |
| T(12*NUM+7) | A |
| T(12*NUM+8) | M |
| T(12*NUM+9) | E |

6. A block in the program or a subprogram must be made to calculate the derivatives. These must be calculated in terms of T block locations since the differential equation subprogram links to this block to compute the derivatives when required. The coding of the derivative calculations may be simplified by equivalence statements if the number of differential equations is invariant.

| FORTRAN name | | |
|---|---|---|
| T(11*NUM) | First derivative | $y'_1$ |
| T(11*NUM-1) | Second derivative | $y'_2$ |
| T(11*NUM-NUM+1) | Last derivative | $y'_n$ |

7. There are three linkages to the differential equation-solving routine, DEDIS, by CALL statements. They are given the names DES, DEI, and DED.

   a. DES    Used once at the beginning to set up the routine for integration of a given set of differential equations. The statement is CALL DES(NUM1, IND, NTEST).

   b. DEI    Linked to the required number of times to integrate all $y_i$ from $t_{initial}$ to $t_{final}$. The CALL statement is CALL DEI(NUM, IND, NTEST).

   c. DED    The statement is CALL DED(NARG).

8. The following fixed point arguments are required for the CALL statements:

   $\left.\begin{array}{l}\text{NUM}\\ \text{NUM1}\end{array}\right\}$ = Number of equations in the system.

   IND = 0, variable interval using Adams-Moulton predictor-corrector method.

   NTEST = Used in the differential equation program in order to have the correct linkage to the differential equation routine. Its value is set in the differential equation routine. It must be the 3rd argument of DES and the argument of the IF statement as indicated below.

NARG = Calculated in the differential equation subprogram and tested in the calling sequence to determine whether an integration step is completed.

EXAMPLE CALLING SEQUENCE

```
   CALL DES(NUM1, IND, NTEST)
 7 IF(NTEST) 15, 15, 11
11 CALL DEI(NUM, IND, NTEST)
15 (Derivative Calculation)
   CALL DED(NARG)
   IF(NARG.EQ.O) GOTO 7
```

SUBPROGRAMS CALLED: None

ERROR RETURNS: If an error is detected, a comment is printed and exit is made to the monitor. Checks are made for the following errors:

| Error | Error comment |
|---|---|
| 1. The number of differential equations N = 0. | Differential equation subroutine DES input error N = 0. |
| 2. N is floating or n > 4096 | Differential equation subroutine DES input error N FLT. |
| 3. Step size H = 0 | Differential equation subroutine DES input error H = 0. |
| 4. Integration mode indicator ≠ 0, 1, or 2. | Differential equation subroutine DES input error IND. |
| 5. Predictor-corrector tolerance E = 0 | Differential equation subroutine DES input error E = 0. |

RESTRICTIONS: See above, ERROR RETURNS

STORAGE:

1. DED $46_{10} = 56_8$
2. DEDIS $1309_{10} = 2435_8$
3. DEI $37_{10} = 45_8$
4. DES $36_{10} = 44_8$

## 5.4 Appendix D — Tape Formats

### Format 1 — Geometry definition tape format

This tape, written by the Geometry Definition section and read by the Transformation section, contains 18 binary records in the first file. Odd-numbered records have 10 fixed-point words; even-numbered records have a variable number of floating-point words. The tape symbol is LA in subroutine OPCAMI, LTAPE in subroutine DEFEN1 (which writes the tape), and LA in subroutine TFLAT (which reads the tape).

Record 1: Body Stations

| | | |
|---|---|---|
| Word | 1 | = 1 |
| | 2 | Number of words in record 2. |
| | 3 | Code* (appears after record 18, at the end of this section). |
| | 4 - 10 | Not used. |

Record 2: Body Stations

| | | |
|---|---|---|
| Word | 1 | First body station. |
| | 2 | Second body station. |
| | . | |
| | . | |
| | . | |
| | n | Last body station. |

Record 3: Basic Body Meridian Lines

| | | |
|---|---|---|
| Word | 1 | = 3 |
| | 2 | Number of words in record 4. |
| | 3 | Code*. |
| | 4 | Number of meridian lines. |
| | 5 - 10 | Not used. |

Record 4: Basic Body Meridian Lines

| | | | |
|---|---|---|---|
| Word | 1 | $\theta$ | First meridian line. |
| | 2 | Number of points in meridian line. | |
| | 3 | Word number of first point in meridian line. | |

Repeat for each additional meridian line.

| | | |
|---|---|---|
| i | $X_1$ | Coordinates of first point in first meridian line. |
| i + 1 | $Y_1$ | |
| i + 2 | $Z_1$ | |

Repeat for each additional point in first meridian line and continue this way for each additional meridian line.

Record 5: Wing Planform (Leading and Trailing Edge Points for Each Control Airfoil or Chord)

| | | |
|---|---|---|
| Word | 1 | = 5 |
| | 2 | Number of words in record 6. |
| | 3 | Code*. |
| | 4 | Number of control airfoils or chords. |
| | 5 - 10 | Not used. |

Record 6: Wing Planform (Leading and Trailing Edge Points for Each Control Airfoil or Chord).

| | | | |
|---|---|---|---|
| Word | 1 | $X_1$ | Coordinates of leading-edge point of first control airfoil or chord. |
| | 2 | $Y_1$ | |
| | 3 | $X_2$ | Coordinates of trailing-edge point of first control airfoil or chord. |
| | 4 | $Y_2$ | |

Follow with leading- and trailing-edge points of remaining control airfoils or chords.

Record 7: Basic Upper Wing Percent Chord Lines

| | | |
|---|---|---|
| Word | 1 | = 7 |
| | 2 | Number of words in record 8. |
| | 3 | Code*. |
| | 4 | Number of percent chord lines. |
| | 5 - 10 | Not used. |

Record 8: Basic Upper Wing Percent Chord Lines

| | | | |
|---|---|---|---|
| Word | 1 | Percent. | First percent chord line. |
| | 2 | Number of points in percent chord line. | |
| | 3 | Word number of first point in percent chord line. | |

Repeat for each additional percent chord line and follow with the coordinates of each point in each percent chord line.

NOTE: The format of record 8 is identical to that of record 4 except that words 1, 4, 7, ..., 3n - 2 contain percent instead of $\theta$, where n = number of percent chord lines.

Record 9: Basic Lower Wing Percent Chord Lines

| | | |
|---|---|---|
| Word | 1 | = 9 |
| | 2 | Number of words in record 10. |
| | 3 | Code*. |
| | 4 | Number of percent chord lines. |
| | 5 - 10 | Not used. |

Record 10: Basic Lower Wing Percent Chord Lines

Same format as record 8.

Record 11: Upper Wing-Body Intersection Points

| | | |
|---|---|---|
| Word | 1 | = 11 |
| | 2 | Number of words in record 12. |
| | 3 | Code*. |
| | 4 | Number of intersection points of upper wing percent chord lines with body. |
| | 5 - 10 | Not used. |

Record 12: Upper Wing-Body Intersection Points

| | | | |
|---|---|---|---|
| Word | 1 | $X_1$ | Coordinates of the intersection of the first upper wing percent chord line with the body. |
| | 2 | $Y_1$ | |
| | 3 | $Z_1$ | |

Repeat for each additional intersection point of the upper wing.

Record 13: Upper Wing Percent Chord Lines (Outside Body)

Word 1 = 13

2 Number of words in record 14.

3 Code*.

4 Number of percent chord lines.

5 - 10 Not used.

Record 14: Upper Wing Percent Chord Lines (Outside Body)

Same format as record 8.

Record 15: Lower Wing-Body Intersection Points

Word 1 = 15

2 Number of words in record 16.

3 Code*.

4 Number of intersection points of lower wing chord percent lines with body.

5 - 10 Not used.

Record 16: Lower Wing-Body Intersection Points

Word 1 $X_1$

2 $Y_1$

3 $Z_1$

Coordinates of the intersection of the first lower wing percent chord line with the body.

Repeat for each additional intersection point of the lower wing.

Record 17: Lower Wing Percent Chord Lines (Outside Body)

Word 1 = 17

2 Number of words in record 18.

3 Code*.

4 Number of percent chord lines

5 - 10 Not used.

Record 18: Lower Wing Percent Chord Lines (Outside Body)

Same format as record 8.

---

* CODE = 0 If this and the next record have been successfully processed.

< 0 An error occurred and the next record is a dummy (all zeros).

> 0 This option was not requested; the next record is a dummy (all zeros).

Format 2 — Geometry transformation tape format

This tape, written by the Geometry Transformation section and read by the Geometry Paneling section of the program, contains 6 binary records in the first file. Odd-numbered records have 10 fixed-point words; even-numbered records have a variable number of floating-point words. The tape symbol is LD in subroutine OPCAMI, LD in subroutine TFLAT (which writes the tape), and ND4 in subroutines INPUTB and INPUTW (which read the tape).

Record 1: Body Meridian Lines

| | | |
|---|---|---|
| Word | 1 | = 1 |
| | 2 | Number of words in record 2. |
| | 3 | Code* (appears after record 6, at the end of this section). |
| | 4 | Number of meridian lines. |
| | 5 - 10 | Not used. |

Record 2: Body Meridian Lines

| | | | |
|---|---|---|---|
| Word | 1 | $\theta$ | First meridian line. |
| | 2 | Number of points in meridian line. | |
| | 3 | Word number of first point in meridian line | |
| | | Repeat for each additional meridian line. | |
| | i | $X_1$ | Coordinates of first point in first meridian line. |
| | i + 1 | $Y_1$ | |
| | i + 2 | $Z_1$ | |
| | | Repeat for each additional point in first meridian line and continue this way for each additional meridian line. | |

Record 3: Wing Percent Chord Lines

| | | |
|---|---|---|
| Word | 1 | = 3 |
| | 2 | Number of words in record 4. |
| | 3 | Code*. |
| | 4 | Number of percent chord lines. |
| | 5 - 10 | Not used. |

Record 4: Wing Percent Chord Lines

| | | | |
|---|---|---|---|
| Word | 1 | Percent. | First percent chord line. |
| | 2 | Number of points in percent chord line. | |
| | 3 | Word number of first point in percent chord line. | |

Repeat for each additional percent chord line and follow with the coordinates of each point in each percent chord line.

NOTE: The format of record 4 is identical to that of record 2 except that words 1, 4, 7, ..., 3n - 2 contain percent instead of $\theta$, where n = number of percent chord lines.

Record 5: Wing-Body Intersection Points

| | | |
|---|---|---|
| Word | 1 | = 5 |
| | 2 | Number of words in record 6. |
| | 3 | Code*. |
| | 4 | Number of intersection points of wing percent chord lines with body. |
| | 5 - 10 | Not used. |

Record 6: Wing-Body Intersection Points

| | | | |
|---|---|---|---|
| Word | 1 | $X_1$ | Coordinates of the intersection of the first wing percent chord line with the body. |
| | 2 | $Y_1$ | |
| | 3 | $Z_1$ | |

Repeat for each additional intersection point.

---

| | | |
|---|---|---|
| * CODE | = 0 | If this and the next record have been successfully processed. |
| | < 0 | An error occurred and the next record is a dummy (all zeros). |
| | > 0 | This option was not requested; the next record is a dummy (all zeros). |

Format 3 — Transformation data tape format

This tape, containing three records in the first file, is written by the Geometry Transformation section of the program; both the Geometry Paneling and Aerodynamics sections read the tape. The tape symbol is LB in subroutine OPCAMI, LB in subroutine TFLAT (which writes the tape), ND1 in subroutine INPUTB (which reads only the first record), and NTAPEB in subroutine INTAPE (which reads all records).

Record 1

| | | | |
|---|---|---|---|
| Word | 1 | x | Components of a unit vector (in original coordinate system) along the x axis of the new coordinate system. |
| | 2 | y | |
| | 3 | z | |
| | 4 | $z_A$ | z coordinate of the transformed and flattened wing. |
| | 5 | $\alpha$ | An original body station X is related to the new body station x by $x = \alpha X + \beta$. |
| | 6 | $\beta$ | |

Record 2

| | | | |
|---|---|---|---|
| Word | 1 | NRX | Number of triplets $(x_i, r_i, z_i)$ in record 3. |
| | 2 | INDEX | For INDEX = j. The triplet $(x_i, r_i, z_i)$ corresponds to the intersection of the leading edge of the wing with the body surface (see record 3). |

Record 3

After the body has been transformed to a new coordinate system such that the new x axis passes through the centers of body ends, the body radius and centroid is found at a number of body stations. Let x = body station, r = body radius, and z = z-coordinate of body centroid.

| | | | |
|---|---|---|---|
| Word | 1 | x | x, r and z at first body station. |
| | 2 | r | |
| | 3 | z | |

Repeat until the record contains NRX triplets.

Format 4 — Geometry Paneling tape format

This tape is written by the Geometry Paneling section for use in the Aerodynamics section. The tape symbol NTAPE2 is used in the Geometry Paneling section and NTAPEC is used in the Aerodynamic section. The sample format given below is that used for a case requiring both body and wing paneling. It should be noted that parts 1, 2, and 3 are deleted for a case that involves only wing paneling and that the tape is not written for a body-alone case. Body paneling data are written in parts 1, 2, and 3 and data for wing paneling in parts 4, 5, and 6; panel corner point coordinates are output in parts 1 and 4 and the additional geometry in parts 2 and 5. One-, two-, and three-records are written in parts 1 and 4 for panels of one, two, and three parts respectively. For these two tape parts, the initial record of each record set always has the format of record 1 as given below and the remaining records of the set have the format described for record 2. A single record is written per panel in parts 2 and 5.

<u>Part 1</u>

Record 1

| | | |
|---|---|---|
| Word | 1 | Body panel number. |
| | 2 | Number of body panel parts. |
| | 3, 4, 5 | Coordinates of inboard leading-edge corner point (first panel part). |
| | 6, 7, 8 | Coordinates of outboard leading-edge corner point (first panel part). |
| | 9, 10, 11 | Coordinates of inboard trailing-edge corner point (first panel part). |

12, 13, 14 — Coordinates of outboard trailing-edge corner point (first panel part).

Record 2

Word 1, 2, 3 — Coordinates of inboard leading-edge corner point (second panel part).

4, 5, 6 — Coordinates of outboard leading-edge corner point (second panel part).

7, 8, 9 — Coordinates of inboard trailing-edge corner point (second panel part).

10, 11, 12 — Coordinates of outboard trailing-edge corner point (second panel part).

These records are repeated for additional body panels as indicated above.

Part 2

Record 1

Word 1 Body panel number.

2, 3, 4 — Coordinates of body panel centroid.

5, 6, 7 — Coordinates of body panel control point.

8 Body panel area.

9 Body panel $\theta$ -inclination angle.

| | |
|---|---|
| 10 | Body panel $\alpha$ -incidence angle. |
| 11 | Body panel (streamwise) chord length. |
| | This single record is repeated for each additional body panel. |

Part 3

Record 1

| | |
|---|---|
| Word 1 | Number of body panels per column. |
| 2 | Fractional value (XPER) used to calculate streamwise location of panel control point (see subroutine PANEL). |
| | This is the only record written for this part. |

Part 4

Record 1

| | |
|---|---|
| Word 1 | Wing panel number. |
| 2 | Number of wing panel parts. |
| 3, 4, 5 | Coordinates of inboard leading-edge corner point (first panel part). |
| 6, 7, 8 | Coordinates of outboard leading-edge corner point (first panel part). |
| 9, 10, 11 | Coordinates of inboard trailing-edge corner point (first panel part). |
| 12, 13, 14 | Coordinates of outboard trailing-edge corner point (first panel part). |

Record 2

| | |
|---|---|
| Word 1, 2, 3 | Coordinates of inboard leading-edge corner point (second panel part). |

4, 5, 6 — Coordinates of outboard leading-edge corner point (second panel part).

7, 8, 9 — Coordinates of inboard trailing-edge corner point (second panel part).

10, 11, 12 — Coordinates of outboard trailing-edge corner point (second panel part).

Record 1 <u>or</u> records 1 <u>and</u> 2 are repeated for additional one- or two-part wing panels as indicated above.

<u>Part 5</u>

Record 1

| | | |
|---|---|---|
| Word | 1 | Wing panel number. |
| | 2, 3, 4 | Coordinates of wing panel centroid. |
| | 5, 6, 7 | Coordinates of wing panel control point. |
| | 8 | Wing panel area. |
| | 9 | Wing panel thickness slope. |
| | 10 | Wing panel camber slope. |
| | 11 | Wing panel (streamwise) chord length. |

This one record is repeated for each additional wing panel.

<u>Part 6</u>

Record 1

| | | |
|---|---|---|
| Word | 1 | Number of wing panels per column. |
| | 2 | Fractional value (XPER) used to calculate streamwise location of panel control point (see subroutine PANEL). |

This is the only record written for this part.

Format 5 — Aerodynamics save tape format

This tape is written in the Aerodynamics section and can be saved for analysis of additional aerodynamic cases on later computer runs. The tape is also used in the Flow Visualization section. The geometrical data of parts 1-4 and the aerodynamic matrices of part 7 are used in the Aerodynamics section. Parts 5, 6, and 8-11 contain data for the Flow Visualization section. The tape symbol is NTAPEC in the Aerodynamics section and N8 in the Flow Visualization section. The sample format given below is that used for a wing-body configuration. Dummy files are written as necessary for wing-alone or body-alone configuration.

Part 1

| | |
|---|---|
| Word 1 | Panel number |
| 2, 3, 4 | x -, y -, and z-coordinates of panel centroid. |
| 5, 6, 7 | x -, y -, and z-coordinates of panel control point. |
| 8 | Panel area. |
| 9 | Panel $\Theta$ —inclination angle. |
| 10 | Panel $\alpha$ —incidence angle. |
| 11 | Panel chord length. |

Words 1 through 11 are repeated for each panel.

| | |
|---|---|
| N + 1 | Code = 1, for wing-alone or body-alone configuration<br>= 2, for wing-body configuration |
| N + 2 | Number of panel rows on body if body-alone or wing-body configuration. Number of panel rows on wing if wing-alone configuration. |
| N + 3 | Number of panel rows on wing if wing-body configuration. Otherwise, = 0. |
| N + 4 | Fractional value for chordwise location of panel control point. |

Part 2

| | |
|---|---|
| Word 1 | Wing panel $\alpha_T$ thickness slope. |

This word is repeated for each wing panel.

Part 3

| | |
|---|---|
| Word 1 | Wing panel $\alpha_C$ camber slope. |

This word is repeated for each wing panel.

Part 4

Record 1

| | |
|---|---|
| Word 1 | Number of body meridian lines. |
| 2 | Angle of first body meridian line. |

Word 2 is repeated for each body meridian line.

Record 2

| | |
|---|---|
| Word 1 | Body radius at first body station. |

This word is repeated for each body station

Record 3

| | |
|---|---|
| Word 1 | Body camber at first body station. |

This word is repeated for each body station.

Record 4 (Same as record 3)

Part 5

| | |
|---|---|
| Word 1 | Code = 1, if wing-alone configuration.<br>= 2, if body-alone configuration.<br>= 3, if wing-body configuration. |
| 2 | Total number of panels on configuration. |
| 3 | Number of body panels. |
| 4 | Number of wing panels. |
| 5 | Number of body stations. |
| 6 | Number of wing panels plus number of panel columns on wing. |
| 7 | Number of panel rows on body if body-alone or wing-body configuration.<br>Number of panel rows on wing if wing-alone configuration. |
| 8 | Number of panel rows on wing if wing-body configuration. Otherwise, = 0. |
| 9 | Mach number. |
| 10 | Symmetry condition. |

Part 6

Record 1

Word 1 — Total number of panels on configuration.

Record 2

Word 1 — Number of panel parts for first panel.

Record 3

| | |
|---|---|
| Word 1, 2, 3 | x- , y- and z-coordinates of inboard leading edge corner point of first panel part. |
| 4, 5, 6 | x- , y- and z-coordinates of outboard leading edge corner point of first panel part. |
| 7, 8, 9 | x- , y- and z-coordinates of inboard trailing edge corner point of first panel part. |
| 10, 11, 12 | x- , y- and z-coordinates of outboard trailing edge corner point of first panel part. |

Record 3 is repeated for each panel part. Record 2 and Record(s) 3 are repeated as a set for each panel.

Record 4

Word 1, 2, 3 — x- , y- and z-coordinates of panel centroid.

Words 1-3 are repeated for each panel.

Record 5

Word 1, 2, 3 — x- , y- and z-coordinates of panel control point.

Words 1 through 3 are repeated for each panel.

Record 6

| | |
|---|---|
| Word 1 | Panel $\alpha$ – incidence angle. |
| 2 | Panel $\Theta$ – inclination angle. |
| 3 | Panel chord length. |

Words 1 through 3 are repeated for each panel.

Part 7

| | | |
|---|---|---|
| Record 1 | $\left[A_{WW}\right]$ | Matrix of aerodynamic influence coefficients due to wing sources. |
| Record 2 | $\left[WW\right]$ | Drag minimization matrix. |
| Record 3 | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift. |
| Record 4 | $\left[WW\right]^{-1}$ | Inverse of drag minimization matrix constrained for wing lift and pitching moment. |
| Record 5 | $\left[U_{WW}\right]$ $\left[V_{WW}\right]$ $\left[W_{WW}\right]$ | Matrices of velocity components on the wing due to wing sources. |
| Record 6 | $\left[U_{BB}\right]$ $\left[V_{BB}\right]$ $\left[W_{BB}\right]$ | Matrices of velocity components on the body due to body surface vortices. |
| Record 7 | $\left[U_{WB}\right]$ $\left[V_{WB}\right]$ $\left[W_{WB}\right]$ | Matrices of velocity components on the wing due to body surface vortices. |
| Record 8 | $\left[U_{BW}\right]$ $\left[V_{BW}\right]$ $\left[W_{BW}\right]$ | Matrices of velocity components on the body due to wing surface vortices. |
| Record 9 | $\left[U_{WW}\right]$ $\left[V_{WW}\right]$ $\left[W_{WW}\right]$ | Matrices of velocity components on the wing due to wing surface vortices. |

Record 10 $\left[A_R\right]$ "Reduced" aerodynamic matrix.

$$\left[A_R\right] = \left\{ \left[A_{WW}\right] - \left[A_{WB}\right]\left[A_{BB}\right]^{-1}\left[A_{BW}\right] \right\}$$

Record 11 $\left[A_R\right]^{-1}$ Inverse of "reduced" aerodynamic matrix.

Record 12 $\left[A_{BB}\right]^{-1}$ Inverse of matrix of aerodynamic influence coefficients on the body due to body surface vortices.

Record 13 $\left[D\right]$ Product matrix.

$$\left[D\right] = \left[A_{WB}\right] \bullet \left[A_{BW}\right]^{-1}$$

Record 14 $\left[E\right]$ Product matrix.

$$\left[E\right] = \left[A_{BB}\right] \bullet \left[A_{BW}\right]$$

<u>Part 8</u>

Word 1 Code = 0., wing thickness effects not to be included.
= 1., wing thickness effects to be included.

2 Angle of attack of configuration.

<u>Part 9</u>

Record 1

Word 1 Number of wing panels plus number of panel columns on wing.

Record 2

Word 1 Wing panel $\alpha_T$ thickness slope.

Record 2 is repeated for each wing column leading edge and wing panel trailing edge.

<u>Part 10</u>

Record 1

Word 1 Number of body stations.

Record 2

Word 1 x-coordinate at first body station.

2 Body radius at first body station.

Words 1 and 2 are repeated for each body station.

Record 3

| | |
|---|---|
| Word 1 | Strength of quadratically varying source at first body station. |
| 2 | Strength of quadratically varying doublet at first body station. |

Words 1 and 2 are repeated for each body station.

Record 4

| | |
|---|---|
| Word 1 | Strength of linearly varying source at body nose. |
| 2 | Strength of linearly varying doublet at body nose. |

<u>Part 11</u>

Record 1

| | |
|---|---|
| Word 1 | Total number of panels on configuration. |

Record 2

| | |
|---|---|
| Word 1 | Strength of vortex on panel surface. |

Word 1 is repeated for each panel.

## Appendix E — Subroutine Listings

Listings of the subroutines in the following index are given in this appendix.

Subroutine Index

```
      SUBROUTINE AERO
C     CONTROL ROUTINE FOR AERODYNAMIC LINKS
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION DICT(5)
      DATA DICT/6HUSE TA,6HSAVE T,6HCOMPUT,6HFLOW V,6HFLOW V/
      REWIND NTAPEA                                                   1
      REWIND NTAPEB                                                   2
      REWIND NTAPEC                                                   3
      REWIND NTAPED                                                   4
      REWIND NTAPEE                                                   5
      REWIND NTAPEF                                                   6
      IGO=INTURP(DICT,3,NTAPEI,NTAPEO)                                7
      IGO=IGO+1
      GO TO (900,50,80,100),IGO
   50 REWIND NTAPEC                                                   10
      ISAVET=-1
      READ (NTAPEI,5010) DUM                                          11
      DO 60 I=4,5
   60 IF (DUM.EQ.DICT(I)) GO TO 550
      CALL USETAP                                                     21
      GO TO 500
   80 ISAVET=1
      GO TO 150
  100 ISAVET=0
  150 CONTINUE
      CALL AMATE
C     KACE = 1      WING ALONE
C     KACE = 2      BODY ALONE
C     KACE = 3      WING-BODY COMBINATION                             27
      GO TO (200,450,300),KACE
 200  CALL INVW                                                       30
      GO TO 400
 300  CONTINUE
      CALL INVBB                                                      33
      CALL REDUCE                                                     35
      CALL INVRW                                                      37
      CALL PARTV                                                      39
 400  CALL MDMATE                                                     41
  450 CALL SAVTAP(ISAVET)                                             43
  500 CALL FORCES                                                     45
      IGO=INTURP(DICT(4),2,NTAPEI,NTAPEO)                             47
```

```
      IF ([illegible]) [illegible],600,[illegible]                51
  550 CALL [illegible]
      GO TO [illegible]                                           53
  [illegible] [illegible] (TAPE)
  [illegible] IF ([illegible](1)-A,[illegible]) [illegible]       57
  [illegible] CALL [illegible](TAPE(C))
      [illegible]                                                 59
  [illegible] WRITE(NTAPE[illegible],6000)
 6000 FORMAT(1H1,[illegible]DATA ERROR, TAPE STATUS NOT INITIATED)

  999 CONTINUE
      RETURN
 5000 FORMAT ([illegible]10.0)
 5010 FORMAT(12A6)
      END
```

```
                                                                    05/31/67
      ALLIN       -  EFN   SOURCE  STATEMENT  -  IFN(S)  -

      SUBROUTINE ALLIN(NTA)
      DIMENSION X(210,3,4),Y(210,3,4),Z(210,3,4),NRCW(2),NPART(210)
      DIMENSION ALPHAS(210),THETA(210)
      DIMENSION XBB(50),R(50),WT(120)
      DIMENSION T(50),TC(50),SST(210) ,CHORD(210)
      COMMON /FLOV1/KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NRCW,XMACH,SYM
      COMMON /FLOV2/X,Y,Z,NPART,ALPHAS,THETA,XBB,R,WT,T,TC,SST,CHORD
      COMMON /BA/XB(210),YB(210),ZB(210)
      COMMON /BB/XC(210),YC(210),ZC(210)
      COMMON /FLOV3/ T11, TC11
      COMMON /THICK/THKW,ARA,CPCALC,CAMN
      N8=NTA
      REWIND NTA                                                            2
      CALL FSF(4,N8,ER)                                                     3
      READ(N8) KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM         4
      CALL FSF(1,N8,ER)                                                    12
      IF(NWING.EQ.0) GO TO 15
      READ(N8) NPANEL                                                      16
      DO 5 J=1,NPANEL
      READ(NTA) NPART(J)                                                   20
      NPT=NPART(J)
      READ(NTA) ((X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4),M=1,NPT)               23
    5 CONTINUE
C     XC,YC,ZC ARE CENTROID POINTS
C     XB,YB,ZB ARE BOUNDARY POINTS
      READ(N8) (XB(J),YB(J),ZB(J),J=1,NPANEL)                              35
      READ(N8) (XC(J),YC(J),ZC(J),J=1,NPANEL)                              44
      READ(N8) (ALPHAS(J),THETA (J),CHORD(J),J=1,NPANEL)                   53
   15 CONTINUE
      CALL FSF(2,N8,ER)                                                    63
      READ(N8) THKW,ARA                                                    64
      CALL FSF(1,N8,ER)                                                    65
      IF(THKW.EQ.0.) GO TO 25
      READ(N8) NWINC                                                       69
      DO 20 J=1,NWINC
      READ(N8) WT(J)                                                       73
   20 CONTINUE
   25 CONTINUE
      CALL FSF(1,N8,ER)                                                    79
      IF(NBODYS.EQ.0) GO TO 35
      READ(N8) NBODYS                                                      83
      READ(N8) (XBB(J),R(J),J=1,NBODYS)                                    85
      READ(N8)(T(J),TC(J),J=1,NBODYS)                                      93
      READ(N8) T11,TC11                                                   101
   35 CONTINUE
      CALL FSF(1,N8,ER)                                                   103
      IF(NWING.EQ.0) GO TO 30
      READ(N8) NPANEL                                                     107
      READ(N8)(SST(J),J=1,NPANEL)                                         109
   30 CONTINUE
      REWIND N8                                                           116
      RETURN
      END
```

```
      SUBROUTINE ALPHAB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),P(3,50),Q(3,5),LOC(3)
C     SUBROUTINE TO CALCULATE BODY PANEL ALPHA
C     ANGLES
C     DEFINE FUNCTION TO BE USED IN CALCULATIONS
C     ANGL CALCULATES ALPHA INCIDENCE ANGLES USING
C     ARCTANGENT ROUTINE
      ANGL(X1,X2,Y1,Y2,Z1,Z2,A)=ATAN(((Z2-Z1)*COS(A)-
     1(Y2-Y1)*SIN(A))/(X2-X1))
C     CALCULATE PANEL ALPHA ANGLES
 80   DO 90 I=1,NPER1                                                    ,1
      DO 87 J=1,NPLN1                                                    ,2
      KGD=KPANEL(I,J)                                                    ,3
      GO TO (83,85,83,85),KGD                                            ,4
 83   ALPHA(I,J)=ANGL(XCOR(I,J),XINT(I,J,1),YCOR(I,J),YINT(I,J,1),
     1ZCOR(I,J),ZINT(I,J,1),THETA(I,J))                                  ,5
      GO TO 87                                                           ,6
 85   ALPHA(I,J)=ANGL(XINT(I,J,1),XCOR(I,J+1),YINT(I,J,1),YCOR(I,J+1),
     1ZINT(I,J,1),ZCOR(I,J+1),THETA(I,J))                                ,7
 87   CONTINUE                                                           ,8      ,9
   90 CONTINUE                                                           ,10     ,11
 100  RETURN                                                             ,12
       END                                                               ,13
```

```
                                                              05/22/67
      AMATE       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE AMATE

C     CONTROL ROUTINE FOR COMPUTING AERODYNAMIC INFLUENCE COEFFICIENTS
C     MATRIX AND VELOCITY COMPONENTS

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      COMMON /BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     C,CHORD(210)
     I,ISYM
     N,NPART(210),NPANEL,NROW(2)
     T,THETA(210),TAIL
     U,U(210)
     V,V(210),VPM(210),VV(210),VPMM(210)
     W,W(210),WPM(210),WW(210),WPMM(210)
     X,X(210,3,4),XBAR(210),XC(210)
     Y,Y(210,3,4),YBAR(210),YC(210)
     Z,Z(210,3,4),ZBAR(210),ZC(210)


C     READ IN GEOMETRICAL DATA
      CALL INTAPE

C     KACE = 1      WING ALONE
C     KACE = 2      BODY ALONE                                        2
C     KACE = 3      WING-BODY COMBINATION
      GO TO (200,300,200),KACE

C     COMPUTE VELOCITY COMPONENTS DUE TO WING SOURCES
C     THICKNESS CASE
  200 NS=NBODY+1
      CALL EVAL(NS,XBAR,1.)

C     COMPUTE VELOCITY COMPONENTS TO WING PANEL SINGULARITIES          6
C     LIFTING CASE
  250 NS=1
      CALL EVAL (NS,XC,0.)
                                                                      9
  300 REWIND NTAPEC                                                   10
      REWIND NTAPEA                                                   11
      REWIND NTAPEB                                                   12
      RETURN
      END
```

```
      AREAP                                                                  08/09/65
         EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE AREAP(X,Y,Z,S,R,NC,NR)
      DIMENSION X(16,16),Y(16,16),Z(16,16),S(15,15),R(15,15)
      DIMENSION A(3),B(3),V(3)
C     SUBROUTINE TO CALCULATE BODY AND WING PANEL
C     AREAS
C     CALCULATE PANEL AREAS
 50   DO 60 I=1,NC                                                  ,1
      DO 60 J=1,NR                                                  ,2
      A(1)=X(I,J)-X(I,J+1)                                          ,3
      A(2)=Y(I,J)-Y(I,J+1)                                          ,4
      A(3)=Z(I,J)-Z(I,J+1)                                          ,5
      B(1)=X(I+1,J+1)-X(I,J+1)                                      ,6
      B(2)=Y(I+1,J+1)-Y(I,J+1)                                      ,7
      B(3)=Z(I+1,J+1)-Z(I,J+1)                                      ,8
      CALL VCROS(A,B,V,D,K)                                         ,9
      S1=D/2.                                                       ,10
      A(1)=X(I,J)-X(I+1,J)                                          ,11
      A(2)=Y(I,J)-Y(I+1,J)                                          ,12
      A(3)=Z(I,J)-Z(I+1,J)                                          ,13
      B(1)=X(I+1,J+1)-X(I+1,J)                                      ,14
      B(2)=Y(I+1,J+1)-Y(I+1,J)                                      ,15
      B(3)=Z(I+1,J+1)-Z(I+1,J)                                      ,16
      CALL VCROS(A,B,V,D,K)                                         ,17
      S2=D/2.                                                       ,18
      S(I,J)=S1+S2                                                  ,19
 60   R(I,J)=S1/S(I,J)                                              ,20    ,21    ,22
 100  RETURN                                                        ,23
       END                                                          ,24
```

```
     ARFREE                                                          07/31/65
    7094 RELMOD ASSEMBLY.

         $IBLDR ARFREE                                                                   ARFR0000
    ASSEMBLED TEXT.

         $TEXT  ARFREE                                                                   ARFR0001
                                   *ARFREE     MICHAEL SYNGE          6-9247      ASSEMBLED   26 JAN 65
                                   *     TO FORM THE FIRST CODE WORD OF AN EMPTY ARRAY.
                                   *
                                   *               CALLING SEQUENCE
                                   *
                                   *                   BUF(M-2)=ARFRE(M)
                                   *
                                   *               WHERE  BUF(M)  IS THE FIRST CELL OF THE ARRAY
                                   *
                                          ENTRY    ARFRE
                                    IRFRE ENTRY    ARFRE
                                   *
BINARY CARD (NOT PUNCHED)
       00000  0500 60 4 00003  10000  ARFRE CLA*    3,4
       00001  0361 00 0 00006  10001        ACL     =0700000
       00002  0602 00 0 00005  10001        SLW     TEMP
       00003  0500 00 0 00005  10001        CLA     TEMP
       00004  0020 00 4 00001  10000        TRA     1,4
                                   *
       00005  0 00000 0 00000  10000  TEMP
       00006  000000700000     10000        *LORG
                        00000  01111        END
    CONTROL DICTIONARY

         $CDICT ARFREE                                                                   ARFR0002

BINARY CARD (NOT PUNCHED)
           000007000000              PREFACE      START=0,LENGTH=7,TYPE=7094,CMPLX=5
           000004000005
           215126512525        ARFREE DECK        LOC=0,LENGTH=7
           000007000000
           215126512560        ARFRE  REAL        LOC=0,LENGTH=0
           000000000000
           315126512560        IRFRE  REAL        LOC=0,LENGTH=0
           000000300000

         $DKEND ARFREE                                                                   ARFR0003

NO MESSAGES FOR THIS ASSEMBLY
```

ARFREE
SYMBOL REFERENCE DATA

07/31/65

REFERENCES TO DEFINED SYMBOLS.

| CLASS | SYMBOL | VALUE | REFERENCES |
|---|---|---|---|
| | ARFRE | 00000 | |
| LCTR | BLCTR | | |
| QUAL | UNQS | | |
| LCTR | // | | |
| | TEMP | 00005 | 2,3 |

```
  ARFUL                                                              07/31/65
 7C94 RELMOD ASSEMBLY.

      $IBLDR ARFUL                                                                          ARFU0300
ASSEMBLED TEXT.
      $TEXT  ARFUL                                                                          ARFU0301
                                   *ARFUL      MICHAEL SYNGE          6-9247      ASSEMBLED  21 AUG 64
                                   *     TO FORM THE FIRST CODE WORD OF A FULL ARRAY.
                                   *
                                   *            CALLING SEQUENCE
                                   *
                                   *               BUF(M-2)=ARFUL(M)
                                   *
                                   *            WHERE  BUF(M)  IS THE FIRST CELL OF THE ARRAY.
                                   *
                                   *       ENTRY    ARFUL
                                   *
BINARY CARD (NOT PUNCHED)
     00000  0500 60 4 00003  10000  ARFUL CLA*    3,4
     00001  0361 00 0 00006  10001        ACL     =0700000700000
     00002  0602 00 0 00005  10001        SLW     TEMP
     00003  0500 00 0 00005  10001        CLA     TEMP
     00004  0020 00 4 00001  10000        TRA     1,4
     00005  0 00000 0 00000  10000  TEMP
     00006  7C0000700000     10000        *LORG
                     00000   01111        END
   CONTROL DICTIONARY

      $CDICT ARFUL                                                                          ARFU0302

BINARY CARD (NOT PUNCHED)
          000007000000              PREFACE    START=0,LENGTH=7,TYPE=7094,CMPLX=5
          000004000005
          215126644360        ARFUL DECK       LOC=0,LENGTH=7
          000007000000
          215126644360        ARFUL REAL       LOC=0,LENGTH=0
          000000000000

      $DKEND ARFUL                                                                          ARFU0003

NO MESSAGES FOR THIS ASSEMBLY
        SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

 CLASS  SYMBOL   VALUE   REFERENCES

        ARFUL    00000
 LCTR    BLCTR
 QUAL    UNQS
 LCTR   //
        TEMP     00005    2,3
```

```
       ARSECR                                                        07/31/65
      7C94 RELMOD ASSEMBLY.

              $IBLDR ARSECR                                                                        ARSE0000
      ASSEMBLED TEXT.
                                                                                                   ARSE0001
              $TEXT  ARSECR
                                      *ARSECR     MICHAEL SYNGE           6-9247     ASSEMBLED   26 JAN 65
                                      *     TO FORM THE SECOND CODE WORD OF A FULL ARRAY.
                                      *
                                      *              CALLING SEQUENCE
                                      *
                                      *                 BUF(M-1)=ARSEC(N,NAME)
                                      *
                                      *              WHERE     N    = THE LENGTH OF THE ARRAY
                                      *                        NAME = THE NAME OF THE ARRAY
                                      *
                                               ENTRY    ARSEC
                                        IRSEC  ENTRY    ARSEC
                                      *
BINARY CARD (NOT PUNCHED)
        00000   0500 60 4 00003   10000   ARSEC CLA*     3,4
        00001   0767 00 0 00022   10000         ALS      18
        00002   0600 00 0 00010   10001         STZ      TEMP
        00003   0622 00 0 00010   10001         STD      TEMP
        00004   0500 00 4 00004   10000         CLA      4,4
        00005   0621 00 0 00010   10001         STA      TEMP
        00006   0500 00 0 00010   10001         CLA      TEMP
        00007   0020 00 4 00001   10000         TRA      1,4
                                          *
        00010   0 00000 0 00000   10000   TEMP
                          00000   01111         END
          CONTROL DICTIONARY

              $CDICT ARSECR                                                                        ARSE0002

BINARY CARD (NOT PUNCHED)
                000011000000              PREFACE     START=0,LENGTH=9,TYPE=7094,CMPLX=5
                000004000005
                215162252351         ARSECR DECK      LOC=0,LENGTH=9
                000011000000
                215162252360         ARSEC  REAL      LOC=0,LENGTH=0
                000000000000
                315162252360         IRSEC  REAL      LOC=0,LENGTH=0
                000000000000

              $DKEND ARSECR                                                                        ARSE0003

NO MESSAGES FOR THIS ASSEMBLY
```

ARSECP
SYMBOL REFERENCE DATA

07/31/65

REFERENCES TO DEFINED SYMBOLS.

| CLASS | SYMBOL | VALUE | REFERENCES |
|---|---|---|---|
| | ARSEC | 00000 | |
| LCTR | BLCTR | | |
| QUAL | UNOS | | |
| LCTR | // | | |
| | TEMP | 00010 | 2,3,5,6 |

```
      SUBROUTINE BCAM(NBODYS,NROWB,XB,ZDELTA,XC,DZDXB,ACB)
      DIMENSION XB(1),ZDELTA(1),XC(1),DZDXB(1),ACB(1)
C     COMPUTES BODY CAMBER SLOPES, GIVEN BODY CAMBER SHAPE
      NB=NBODYS-1
      DO 100 I=1,NB
  100 DZDXB(I+1)=(ZDELTA(I+1)-ZDELTA(I))/(XB(I+1)-XB(I))
      DZDXB(1)=DZDXB(2)
      DO 120 J=1,NROWB
      DO 110 I=1,NB
      IF (XB(I)-XC(J)) 110,115,115
  110 CONTINUE
  115 ACB(J)=DZDXB(I-1)+(XC(J)-XB(I-1))/(XB(I)-XB(I-1))*(DZDXB(I)-DZDXB(
     1I-1))
  120 CONTINUE
      RETURN
      END
```

```
       BCUTX                                                        07/31/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE BCUTX(B,N,X,EP,NA,A)
C
      DIMENSION B(1),C(4),A(2,1),LQ(2),Q(3)
C
C     USED BY WBXC TO CUT THE BODY WITH A STATION PLANE
C     TO GET A 2-DIM. ARRAY OF PTS. (A) IN THE PLANE.
C         A = (Y1,Z1,Y2,Z2,.....)
C     NA = NO. PTS. IN A.
C     INPUTS ARE B,N,X,EP.  OUTPUTS ARE NA,A
      DATA C/-1.,3*0./
C
      J=0                                                          ,1
      C(4)=X                                                       ,2
 100  DO 1000 K=1,N                                                ,3
      K1=3*K                                                       ,4
      K2=B(K1-1)                                                   ,5
      K3=B(K1)                                                     ,6
      CALL PCLXN(C,B(K3),3,K2,EP,1,Q,LQ,L)                         ,7
      IF (L .NE. 1) GO TO 1000                                     ,8    ,9    ,10
      J=J+1                                                        ,11
      A(1,J)=Q(2)                                                  ,12
      A(2,J)=Q(3)
C                                                                  ,13
 1000 CONTINUE                                                     ,14   ,15
      NA=J                                                         ,16
 9000 RETURN                                                       ,17
      END                                                          ,18
```

```
      SUBROUTINE BITURP(X,Y,IXY,NXY,KD,XI,YI,NU)
      DIMENSION X(1),Y(1),NU(3),C(4),YIQ(2)
C
C     GIVEN NXY POINTS AND X=XI,INTERPOLATE TO FIND Y=YI.
C     X AND Y VALUES SPACED IXY APART.  IXY=1 IF CONSECUTIVE.
C     KD=1 IF LINEAR INTERPOLATION, KD=2 IF BIQUADRATIC.
C
      LOGICAL  LINT,LIF,LIL
C
      NU=0                                                        ,1
      LINT=.FALSE.                                                ,2
      LIF=.FALSE.                                                 ,3
      LIL=.FALSE.
C                                                                 ,4
 100  IF (IXY)110,110,200                                         ,5
 110  NU=-1                                                       ,6
      GO TO 5000                                                  ,7
 200  IF (NXY-2)210,220,300                                       ,8
 210  NU=-2                                                       ,9
      GO TO 3200
C     USE LINEAR INTERPOLATION IF ONLY 2 POINTS.                  ,10
 220  LINT=.TRUE.
C                                                                 ,11
 300  IX=IXY*(NXY-1)+1                                            ,12
      JXY=IXY
C                                                                 ,13
 400  IF (X(IX)-X)420,410,430                                     ,14
 410  NU=-3                                                       ,15
      GO TO 3200
C
C     X-VALUES OF POINTS IN DECREASING ORDER.
C     WE WILL SEARCH FROM RIGHT TO LEFT INSTEAD OF
C       LEFT TO RIGHT.                                            ,16
 420  IA=IX+IXY                                                   ,17
      JXY=-JXY                                                    ,18
      GO TO 500
C                                                                 ,19
 430  IA=1-IXY
C     SEARCH FOR INTERVAL CONTAINING XI                           ,20
 500  DO 600 I=1,NXY                                              ,21
      IA=IA+JXY                                                   ,22
 510  IF (XI-X(IA))700,800,600                                    ,23
 600  CONTINUE                                                    ,24    ,25
 610  NU=2                                                        ,26
      GO TO 800
C                                                                 ,27
 700  IF (I-2)710,900,1000                                        ,28
 710  NU=1
C
C     XI FALLS EXACTLY ON A POINT OR IS OUTSIDE THE TABLE.        ,29
 800  YI=Y(IA)                                                    ,30
      GO TO 3100
C
```

```
C     XI IN 1ST INTERVAL (OR LAST IF GOING FROM RIGHT)                  ,31
 900  LIF=.TRUE.
C                                                                       ,32
 1000 IF (I-NXY)1100,1010,1010
C     XI IN LAST INTERVAL (OR 1ST)                                      ,33
 1010 LIL=.TRUE.
C                                                                       ,34
 1100 I2=IA-JXY                                                         ,35
 1200 IF (KD-2)3000,1250,1250                                           ,36
 1250 IF (LINT) GO TO 3000
C
C     QUADRATIC OR BIQUADRATIC INTERPOLATION                            ,37   ,38   ,39
 1300 JA=IA                                                             ,40
 1400 DO 1800 IZ=1,2                                                    ,41
      JA=JA-JXY                                                         ,42
 1500 IF ((IZ.EQ.1 .AND.LIL) .OR. (IZ.EQ.2 .AND.LIF))GO TO 1800         ,43   ,44   ,45
 1600 CALL QRAT(X(JA),Y(JA),JXY,1,XI,YIQ(IZ),C,MU)                      ,46
      YI=YIQ(IZ)                                                        ,47
 1700 IF (MU)1710,1800,1710                                             ,48
 1710 NU=IZ+2
C     TWO POINTS MUST HAVE SAME X                                       ,49
      GO TO 3000
C                                                                       ,50
 1800 CONTINUE
C                                                                       ,51   ,52
 1900 IF (LIF.OR.LIL) GO TO 9000
C     BIQUADRATIC INTERPOLATION                                         ,53   ,54   ,55
 2000 KXY=JXY+JXY                                                       ,56
      I1=IA-KXY                                                         ,57
      I4=IA+JXY                                                         ,58
      YI=((X(IA)-XI)*YIQ+(XI-X(I2))*YIQ(2))/(X(IA)-X(I2))               ,59
      GO TO 9000
C
C     LINEAR INTERPOLATION                                              ,60
 3000 YI=Y(I2)+(XI-X(I2))*((Y(IA)-Y(I2))/(X(IA)-X(I2)))
C                                                                       ,61
 3100 IF (NU)3200,9000,3200
C
C     ERROR.  REDUCE ERROR COUNTER AND WRITE MESSAGE                    ,62
 3200 LTAPE=NU(2)                                                       ,63
      NU(3)=NU(3)-1                                                     ,64
 3300 IF (NU(3).LE.0 .OR. LTAPE.LE.0) GO TO 9000                        ,65   ,66   ,67
 3400 WRITE (LTAPE,3410)NU(1),XI,YI,NXY,(X(I),Y(I),I=1,IX,IXY)          ,68   ,69   ,70   ,71   ,72   ,73

 3410 FORMAT(13H0BITURP ERROR I3,8H FOR X = F14.6,1H.14H  RETURNED Y =
     .F14.6,I6,17H POINTS (X,Y) ARE/(1H 8F14.6))
C
 9000 RETURN                                                            ,74
      END                                                               ,75
```

```
      SUBROUTINE BJTURP(X,Y,IXY,NXY,KD,XI,YI,NU)
      DIMENSION X(1),Y(1),NU(3),C(4),YIQ(2)
C
C     GIVEN NXY POINTS AND X=XI,INTERPOLATE TO FIND Y=YI.
C     X AND Y VALUES SPACED IXY APART.  IXY=1 IF CONSECUTIVE.
C     KD=1 IF LINEAR INTERPOLATION, KD=2 IF BIQUADRATIC.
C
      LOGICAL  LINT,LIF,LIL
C
      NU=0
      LINT=.FALSE.
      LIF=.FALSE.
      LIL=.FALSE.
C
  100 IF (IXY)110,110,200
  110 NU=-1
      GO TO 9000
  200 IF (NXY-2)210,220,300
  210 NU=-2
      GO TO 3200
C     USE LINEAR INTERPOLATION IF ONLY 2 POINTS.
  220 LINT=.TRUE.
C
  300 IX=IXY*(NXY-1)+1
      JXY=IXY
C
  400 IF (X(IX)-X)420,410,430
  410 NU=-3
      GO TO 3200
C
C     X-VALUES OF POINTS IN DECREASING ORDER.
C     WE WILL SEARCH FROM RIGHT TO LEFT INSTEAD OF
C       LEFT TO RIGHT.
  420 IA=IX+IXY
      JXY=-JXY
      GO TO 500
C
  430 IA=1-IXY
C     SEARCH FOR INTERVAL CONTAINING XI
  500 DO 600 I=1,NXY
      IA=IA+JXY
  510 IF (XI-X(IA))700,800,600
  600 CONTINUE
  610 NU=2
      GO TO 800
C
  700 IF (I-2)710,900,1000
  710 NU=1
C
C     XI FALLS EXACTLY ON A POINT OR IS OUTSIDE THE TABLE.
  800 YI=Y(IA)
      GO TO 3100
C
```

```
C       XI IN 1ST INTERVAL (OR LAST IF GOING FROM RIGHT)
  900   LIF=.TRUE.
C
 1000 IF (I-NXY)1100,1010,1010
C       XI IN LAST INTERVAL (OR 1ST)
 1010 LIL=.TRUE.
C
 1100 I2=IA-JXY
 1200 IF (KD-2)3000,1250,1250
 1250 IF (LINT) GO TO 3000
C
C       QUADRATIC OR BIQUADRATIC INTERPOLATION
 1300 JA=IA
 1400 DO 1800 IZ=1,2
      JA=JA-JXY
 1500 IF ((IZ.EQ.1 .AND.LIL) .OR. (IZ.EQ.2 .AND.LIF))GO TO 1800
 1600 CALL QRAT(X(JA),Y(JA),JXY,1,XI,YIQ(IZ),C,MU)                  62
      YI=YIQ(IZ)
 1700 IF (MU)1710,1800,1710
 1710 NU=IZ+2
C       TWO POINTS MUST HAVE SAME X
      GO TO 3000
C
 1800 CONTINUE
C
 1900 IF (LIF.OR.LIL) GO TO 9000
C       BIQUADRATIC INTERPOLATION
 2000 KXY=JXY+JXY
      I1=IA-KXY
      I4=IA+JXY
      YI=((X(IA)-XI)*YIQ+(XI-X(I2))*YIQ(2))/(X(IA)-X(I2))
      GO TO 9000
C
C       LINEAR INTERPOLATION
 3000 YI=Y(I2)+(XI-X(I2))*((Y(IA)-Y(I2))/(X(IA)-X(I2)))
C
 3100 IF (NU)3200,9000,3200
C
C       ERROR.  REDUCE ERROR COUNTER AND WRITE MESSAGE
 3200 LTAPE=NU(2)
      NU(3)=NU(3)-1
 3300 IF (NU(3).LE.0 .OR. LTAPE.LE.0) GO TO 9000
 3400 WRITE (LTAPE,3410)NU(1),XI,YI,NXY,(X(I),Y(I),I=1,IX,IXY)     100
 3410 FORMAT(13H0BITURP ERROR I3,8H FOR X = F14.6,1H,14H  RETURNED Y =
     .F14.6,I6,17H POINTS (X,Y) ARE/(1H 8F14.6))
C
 9000 RETURN
      END
```

```
        BLDM                                                          08/16/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE BLDM (NM,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHAM,THETAM
     1,CPM,CL,CD,CM)

C     COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT CN BODY

      DIMENSION AREA(1),XBAR(1),ZBAR(1),ALPHAM(1),THETAM(1),CPM(1)


      DRAG=0.                                                         ,1
      XLIFT=0.                                                        ,2
      XM=0.0
                                                                      ,3
      DO 100 J=1,NM                                                   ,4
      F=CPM(J)*AREA(J)                                                ,5
      XL=-F*COS(THETAM(J))                                            ,6
      XD=F*ALPHAM(J)                                                  ,7
      XM=XM-XL*(XBAR(J)-XP)+XD*(ZBAR(J)-ZP)                           ,8
      XLIFT=XLIFT+XL                                                  ,9
      DRAG=DRAG+XD                                                    ,10
  100 CONTINUE
                                                                      ,11    ,12
      CL=XLIFT/RFAREA                                                 ,13
      CD=DRAG/RFAREA                                                  ,14
      CM=XM/RFAREA
                                                                      ,15
      RETURN                                                          ,16
      END                                                             ,17
```

```
      FUNCTION BLUNDR (IX,IY,IZ,MODE,NMAX)
      COMMON DUM(4),N9,DM(3),N5,N6
      BLUNDR=0.
      GO TO(1,2,3,4),MODE
    1 IF(IZ*IY.GT.NMAX) GO TO 5
      RETURN
    2 IF(IZ*IX.GT.NMAX) GO TO 5
      RETURN
    3 IF((IX*IY).GT.NMAX) GO TO 5
      RETURN
    4 IF((IX*IY+IY*IZ+IZ*IX).GT.NMAX) GO TO 5
      RETURN
    5 BLUNDR=1.
      WRITE(N6,100) NMAX
  100 FORMAT(1HO,8(1H*),45HGRIDS INPUT ERROR PRODUCT OF INCREMENT PARAME
     $17HTERS GREATER THAN,2X,I4,40(1H*))
      RETURN
      END
```

16

```
      SUBROUTINE BODCR(B,N,X,EP,CHD,MAXE,S,E,NE,Y,Z,AREA,RAD,NU)
C
C     FINDS THE AREA,EQUIV. RADIUS, AND CENTROID OF A BODY SECTION.
C
      DIMENSION B(1),EP(5),S(2,N),E(MAXE),NU(3),CG(2)
C
C     THE BODY IS GIVEN AS A SET OF MERIDIAN LINES (SEE BODY1).
C       THE BODY IS SYMMETRICAL ABOUT A PLANE PARALLEL TO THE XZ PLANE.
C       WE HAVE THE MERIDIAN LINES FOR HALF THE BODY.
C     THE BODY SECTION IS THE INTERSECTION OF A STATION PLANE WITH THE BODY.
C    INPUTS ARE
C     B = BODY MERIDIAN LINE POINTS (WITH HEADER)
C     N = NO. OF MERIDIAN LINES
C     X = BODY STATION
C     EP = ARRAY OF TOLERANCES
C     CHD = CHORD HEIGHT TOLERANCE USED BY ENRYCH TO DETERMINE
C           SPACING OF INTERPOLATED POINTS BETWEEN MERIDIANS.
C     MAXE = LENGTH OF ARRAY E
C     NU(2) = TAPE NO. FOR ENRYCH ERROR MESSAGE
C     NU(3) = ERROR LIMITER AND COUNTER (ALSO OUTPUT)
C    OUTPUTS ARE
C     S = ARRAY OF N POINTS (Y,Z) IN SECTION
C     E = IS ESSENTIALLY A SCRATCH ARRAY FOR ENRICHED SECTION POINTS.
C         THE 1ST N CELLS ARE THE Y VALUES OF S.
C         THE NEXT N CELLS ARE THE Z VALUES OF S.
C         THE NEXT NE CELLS ARE THE DENSE ARRAY OF POINTS,
C         STORED (Y,Z,Y,Z,....) ON THE SECTION.
C       E IS NOT USED FOR DEGENERATE SECTIONS.
C       E IS USED BUT NO ENRICHING TAKES PLACE IF N .LT. 4 OR
C         IF CHD .LE. 0.
C     NE IS EXPLAINED UNDER E
C     Y = Y COORD. OF CENTROID OF WHOLE SECTION.
C     Z = Z COORD. OF CENTROID
C     AREA = AREA OF WHOLE SECTION
C     RAD = AVERAGE DISTANCE FROM CENTROID TO EACH MERIDIAN LINE.
C     NU = 0 IF SUCCESS
C          1 IF CUT THRU BODY DOES NOT INTERSECT ALL MERIDIAN LINES
C            OR ELSE N .LT. 2
C          2,3,4, OR 5 IF ENRYCH ERROR.
C          6 IF 1ST AND LAST PTS IN SECTION DO NOT HAVE SAME Y WITHIN EP(2)
C
      KEE = 0                                                          ,1
      NU=1
C     CUT BODY WITH PLANE X = X TO GET SECTION S OF NS POINTS          ,2
      CALL BCUTX(B,N,X,EP(1),NS,S)                                     ,3
      IF(NS .NE. N  .OR.  N .LT. 2) GO TO 8000                         ,4    ,5    ,6
      NU=6
C     CHECK THAT 1ST AND LAST PTS IN SECTION HAVE SAME Y VALUE         ,7
      IF(ABS(S(1,1)-S(1,N)) .GT. EP(2)) GO TO 8000                     ,8    ,9    ,10
      Y=(S(1,1)+S(1,N))/2.
C     CHECK FOR SECTION WITH ZERO AREA                                 ,11
      IF(ABS(S(2,1)-S(2,N)) .GT. EP(4)  .AND.  N .GT. 2) GO TO 500     ,12   ,13   ,14
      AREA=0.                                                          ,15
```

```
      GO TO 1500
C
C     PREPARE TO ENRICH SECTION.  WE MUST HAVE Y,Z IN SEPARATE
C     ARRAY SO USE 1ST PART OF E ARRAY                                  ,16
  500 DO 600 I=1,N                                                      ,17
      E(I)=S(1,I)                                                       ,18
      II=I+N                                                            ,19
  600 E(II)=S(2,I)
C                                                                       ,20   ,21
      IE=II+1
C     THE ENRICHED SECTION POINTS WILL START IN E(IE)                   ,22
      NE=MAXE-II-2
C     NE=AVAILABLE STORAGE FOR ENRICHED POINTS.  WE SUBTRACT 2 BECAUSE
C     CEGAR WILL LATER FILL IN 2 CELLS AT END OF ARRAY.                 ,23
      CALL ENRYCH(E,E(N+1),N,CHD,E(IE),NE,NU)                           ,24
      IF(NU .EQ. 0) GO TO 1000                                          ,25   ,26   ,27
      KEE = 1                                                           ,28
      IF(NU .NE. 3) GO TO 8000
C     NOTE WE IGNORE ERROR 3 FROM ENRYCH, ASSUMING THAT CALLING PROGRAM
C     HAS SET NU(2), NU(3) SO THAT ENRYCH PRINTED ERROR MESSAGE.
C     IF ERROR FROM ENRYCH, EXPLANATORY MESSAGE WRITTEN AT END (KEE = 1)

C     FIND AREA, CENTRCID OF ENRICHED SECTION                           ,29   ,30   ,31
 1000 CALL CEGAR(NE/2,E(IE), AREA,CG)                                   ,32
      AREA=AREA+AREA
C     IF AREA TOO SMALL, CG MAY BE INACCURATE                           ,33
      IF(AREA .LE. EP(3)) GO TO 1500                                    ,34   ,35   ,36
      Z=CG(2)                                                           ,37
      GO TO 2000
C                                                                       ,38
 1500 CONTINUE
C     FOR Z OF CENTROID, USE AVERAGE OF 1ST AND LAST POINTS.            ,39
      Z=(S(2,1)+S(2,N))/2.
C                                                                       ,40
      IF(AREA)1600,160C,2000                                            ,41
 1600 RAD=0.                                                            ,42
      GO TO 3000
C                                                                       ,43
 2000 CONTINUE

C     COMPUTE RADIUS OF SECTION                                         ,44
      SUMRAD=0.                                                         ,45
      DO 2500 I=1,NS                                                    ,46
      RR=(S(1,I)-Y)**2  + (S(2,I)-Z)**2                                 ,47
      R=SQRT(RR)                                                        ,48
      SUMRAD = SUMRAD + R                                               ,49
      IF(I .EQ. 1  .OR.  I .EQ. NS) GO TO 2500                          ,50   ,51   ,52
      SUMRAD = SUMRAD + R                                               ,53
 2500 CONTINUE                                                          ,54   ,55
      FNS=NS+NS-2                                                       ,56
      RAD=SUMRAD/FNS                                                    ,57
 3000 CONTINUE                                                          ,58
      NU=0
C                                                                       ,59
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

8000 CONTINUE                                                    ,60
     LTAPE=NU(2)                                                 ,61
     IF(.NOT.(NU .NE. 0  .AND.  LTAPE .GT. 0)) GO TO 9000        ,62  ,63  ,64
     WRITE(LTAPE,8100)NU(1),(J,(S(I,J),I=1,2),J=1,NS)            ,65  ,66  ,67  ,68  ,69  ,70
                                                                 ,71  ,72  ,73  ,74
8100 FORMAT(12H0BODCR ERROR I3,5X
   1 57HFAILURE IN FINDING CENTROID AND RADIUS OF A BODY SECTION
   2 /30H BOCY SECTICN POINTS (Y,Z) ARE/(1H I3,2F20.6))
9000 CONTINUE                                                    ,75
     IF(KEE .EQ. 0  .CR.  LTAPE .LE. 0) GO TO 9900               ,76  ,77  ,78
     WRITE (LTAPE,9100)                                          ,79  ,80
9100 FORMAT(50HOTHE FCUR PTS WRITTEN BY ENRYCH ARE (Y,Z) OR (Z,Y) /
   163H COORD. CF BOCY MERIDIAN LINES, BEFORE OR AFTER TRANSFORMATION,
   2/49H AT A GIVEN STATION.  IF CODE = 3, NO POINTS ARE
   3 27HINTERPOLATED IN AN INTERVAL /
   4 26H BUT THE PROGRAM CONTINUES )
9900 CONTINUE                                                    ,81
     IF(ABS(Z)-EP(5)) 9910,9910,9920                             ,82
9910 Z=0.                                                        ,83
9920 IF(RAD-EP(5)) 9930,9930,9940                                ,84
9930 RAD=0.                                                      ,85
9940 CONTINUE                                                    ,86
     RETURN                                                      ,87
     END                                                         ,88
```

```
           BODY                                                                    08/16/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

       SUBROUTINE BODY
       COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
      1NWRITE,NBODY,NWING,XMACH,SYM,KACE
       COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
      1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
      2,NPLANE,NPLN1
       COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
      1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
      2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
      3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
      4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
      50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
       DIMENSION ND3(10855)
       EQUIVALENCE (ND3(1),NPLNB)
C      CONTROL PROGRAM FOR BODY PANELING SUBROUTINES
       DATA NAME/6HDEFINE/
C      DEFINE CONSTANTS
C      NBODY IS NUMBER OF BODY PANELS
C      KODEB,KODEI ARE SCRATCH TAPE ERROR CODES
C      IF KODEB OR KODEI ARE SET = 1 IN INPUTB
C      THEN TAPE ERROR ENCOUNTERED
 10    NBODY=0                                                          ,1
       KODEB=0                                                          ,2
       KODEI=0
C      INITIALIZE COMMON ARRAYS                                         ,3
       DO 15 I=1,10855                                                  ,4
 15    ND3(I)=0.
C      CALL SUBROUTINES                                                 ,5    ,6
 20    CALL INPUTB                                                      ,7
       IF (KODEB) 150,25,150                                            ,8
 25    IF (KODEI) 150,30,150                                            ,9
 30    CALL CRNRB                                                       ,10
       IF (KODEB) 150,40,80                                             ,11
 40    CALL AREAP(XCOR,YCOR,ZCOR,AREA,ARAT,NPER1,NPLN1)                 ,12
       CALL CENTRD (XCOR,YCOR,ZCOR,ARAT,XCEN,YCEN,ZCEN,
      1NPER1,NPLN1)                                                     ,13
       CALL THETAB                                                      ,14
       IF (KODEB) 150,50,80                                             ,15
 50    CALL CNTRLB                                                      ,16
       IF (KODEB) 150,60,80                                             ,17
 60    CALL ALPHAB                                                      ,18
       IF (KODEB) 150,90,80                                             ,19
 80    WRITE (NWRITE,900)                                               ,20   ,21
 90    CALL OUTPTB                                                      ,22
       IF (KODEB) 150,100,150                                           ,23
 100   RETURN                                                           ,24
 150   READ (NREAD,890) KARD                                             25   ,26   ,27
       IF (KARD-NAME) 150,160,150                                       ,28
 160   BACKSPACE NREAD                                                  ,29
```

```
      KODEB=-1                                                        ,30
      GO TO 100                                                       ,31
  890 FORMAT (A6)
  900 FORMAT (1H1,9X,13HEPROR MESSAGE/10X,
     149HPANELLING DATA NOT COMPLETED DUE TO PROGRAM ERROR/10X,
     238HINCOMPLETE PANELLING DATA TO BE OUTPUT)                      ,32
       END
```

```
C     CALLED BY GEOMD
C     READS A BODY DEFINITION IN THE FORM OF CROSS-SECTIONS AT VARIOUS
C       STATION PLANES, AND THEN COMPUTES BODY MERIDIAN LINES (ALSO CALLED
C       PERCENT LINES) EACH OF WHICH CONSISTS OF 3-DIMENSIONAL POINTS ON THE
C       BODY SURFACE.
C
      SUBROUTINE BODY1(DAT,LI,LO,B,LSTA,NSTA,LBPL,MBPL,NBPL
     1,LTHETS,AXIS,TITLE,LGDEF,NU)
C
      DIMENSION DAT(2),B(1),TITLE(12),NU(3),AXIS(2)
      DATA   MAX,SR/150,5HBODY1/
      LOGICAL LGDEF(3,6)
C
C     INPUTS ARE DAT,LI,LO,NU(2)
C     INPUT-OUTPUTS ARE B,NU(3)
C     OUTPUTS ARE LSTA,NSTA,LBPL,MBPL,NBPL,LTHETS,AXIS,TITLE,NU(1),LGDEF
C
C     DEFINING STATION ARRAY STARTS IN B(LSTA)
C     NSTA = NO. OF STATIONS
C
C     BODY MERIDIAN LINE ARRAY (WITH HEADER) STARTS IN B(LBPL)
C     MBPL = NO. OF BODY PERCENT LINES (MERIDIAN LINES)
C     NBPL = NO OF CELLS IN ARRAY
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, .........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...   ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     THE STANDARD SET OF THETAS STARTS IN B(LTHETS).
C       THERE ARE MBPL OF THEM
C     AXIS = Y,Z COORD. OF MAIN BODY AXIS
C     TITLE = BODY TITLE
C     LGDEF(1,1) = .TRUE. TO INDICATE BODY DEF. CALLED FOR.
C     LGDEF(2,1) = .TRUE. IF BODY SUCCESSFULLY DEFINED.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C
      LGDEF(1,3)=.TRUE.                                                   ,1
      READ(LI,100)TITLE                                                   ,2    ,3    ,4
  100 FORMAT(12A6)
```

```
          BODY1                                                            09/14/65
        EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)
      WRITE(LO,110)DAT,TITLE                                               ,5    ,6    ,7
  110 FORMAT(16H1BODY DEFINITION 38X 2A6/1H012A6)
C
      READ(LI,200)BNS,BTHETA,AXIS,CHD                                      ,8    ,9    ,10
  200 FORMAT(6F10.0)
      NSTA=BNS                                                             ,11
      NTHETA=BTHETA                                                        ,12
      WRITE(LO,300)NSTA,NTHETA,AXIS,CHD                                    ,13   ,14   ,15
  300 FORMAT(10HOTHERE ARE I3,13H STATIONS AND I3,
     1 15H MERIDIAN LINES/
     2 59HOTHE MAIN BODY AXIS (PARALLEL TO THE X AXIS) IS LOCATED AT
     3/8X,3HY = F9.4,4X,3HZ = F9.4/
     4 26HOCHORD-HEIGHT TOLERANCE = F8.4//)
C
C     CHECK NSTA,NTHETA
      IF(NSTA .LT. 2 .CR. NSTA .GT. MAX .OR. NTHETA .LT. 2 .OR.
     1 NTHETA .GT. MAX)GO TO 8000
C
C     RESERVE STORAGE                                                      ,16   ,17   ,18
      LSTA=IRSERV(NSTA,B,LSTA)                                             ,19
      LTHETS=IRSERV(NTHETA,B,LTHETS)                                       ,20
      LTHETA=IRSERV(NTHETA,B,LTHETA)                                       ,21
      LRHO=IRSERV(NTHETA,B,LRHO)                                           ,22
      NTHET3=3*NTHETA                                                      ,23
      LPTS=IRSERV(NTHET3,B,LPTS)
C     THESE LAST 3 ARRAYS ARE SCRATCH STORAGE FOR BODY1R.
C     LEAVE BALANCE OF STORAGE FOR MERIDIAN LINES.
C       WE WILL RELEASE EXCESS LATER.                                      ,24
      MBPL=NTHETA                                                          ,25
      NBPL=IPACK(B)                                                        ,26
      MU=2
C     SEE IF ENOUGH ROCM                                                   ,27
      IF(NBPL-NTHET3*(NSTA+1))8000,600,600                                 ,28
  600 LBPL=IRSERV(NBPL,B,LBPL)                                             ,29
      IF(LBPL)8000,800C,700
C
C     READ STANDARD SET OF THETAS                                          ,30
  700 LTH=LTHETS+NTHETA-1                                                  ,31
      READ(LI,200)(B(I),I=LTHETS,LTH)
C
C     READ EACH DEFINING SECTION.  FIND X,Y,Z,RHO,THETA FOR EACH PT.
C     WRITE DATA FOR EACH SECTION.  STORE PTS. IN B(LBPL) IN
C       MERIDIAN LINE CRDER.                                               ,32   ,33   ,34   ,35   ,36
      MU=3                                                                 ,37
      CALL BODY1R(LI,LC,NTHETA,NSTA,AXIS,TITLE,DAT,B(LTHETS),
     1 B(LTHETA),B(LRHO),B(LPTS),B(LBPL),B(LSTA),NU)                       ,38
      IF(NU)8000,900,8C00
C
C     PERFORM 3D INTERPOLATION ON EACH MERIDIAN LINE.
C     THE ENRICHED MERIDIAN LINES (PRECEDED BY HEADER) WILL BE
C       STORED STARTING IN B(LBPL).                                        ,39
  900 MAXP=NBPL/3                                                          ,40
      MU=4                                                                 ,41
      CALL RICHNA(B(LBPL),MAXP,NTHETA,NSTA,B(LTHETS),.001
```

```
         BODY1                                                       09/14/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

     1 ,CHD,0,NA,NU)
C      NA = TOTAL NO OF PTS                                           ,42
       IF(NU)8000,1000,8000                                           ,43
 1000  NBPL=3*(NA+NTHETA)
C
C      RELEASE UNUSED STORAGE                                         ,44
       MU=5                                                           ,45
       IF(IRLEASE(B(LBPL),NBPL))8000,200C,8000
C
C      DELETE SCRATCH ARRAYS                                          ,46
 2000  IF(IDLETE(B(LTHETA)))8000,2100,8000                            ,47
 2100  IF(IDLETE(B(LRHO)))8000,2200,8000                              ,48
 2200  IF(IDLETE(B(LPTS)))8000,3000,8000
C
C      WRITE BODY MERIDIAN LINES                                      ,49
 3000  CALL BODY1M(B(LBPL),NTHETA,AXIS,TITLE,DAT,LO,B(LSTA))
C
C      SUCCESS                                                        ,50
 4000  LGDEF(2,3)=.TRUE.                                              ,51
       GO TO 9000
C
C      ERROR.  WRITE MESSAGE.                                         ,52
 8000  I=MERR(NU,NU+1,ML,SR)                                          ,53
       NU=MU
C      IF MU = 3 OR 4, RELEASE BULK OF LBPL STORAGE SO WING CAN BE PROCESSED    ,54
       IF(MU .LT. 3 .OR. MU .GT. 4)GO TO 9000                         ,55    ,56    ,57
       I=IRLEAS(B(LBPL),1000)
C                                                                     ,58
 9000  RETURN                                                         ,59
       END                                                            ,60
```

```
C      USED BY BODY1 TO PRINT BODY MERIDIAN LINES.
C      ALSO COMPUTES RHC, THETA FOR EACH POINT.
C      THE RHO, THETA ORIGIN IS ON BODY AXIS.
C
       SUBROUTINE BODY1P(B,NB,AXIS,TITLE,DAYT,LO,STA)
C
       DIMENSION B(1),AXIS(2),TITLE(12),DAYT(2),STA(1),XYZ(3)
       LOGICAL NEWPCT,WTITLE
       DATA MAXLIN,DEGRAD,BLANK,STAR/45,0206712273406,1H ,3H  */
C
C      B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C      STA IS ARRAY OF BODY STATIONS.  POINTS OF MERIDIAN LINES
C        WHICH HAVE X = A STATION ARE MARKED BY AN *
C      MAXLIN = MAX NO. OF LINES PER PAGE
C      MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C        ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
       MAXEL=3*(MAXLIN-2)                                          ,1
       NELREM=0
C                                                                  ,2
       DO 4000 K=1,NB
C      WORKING ON KTH MERIDIAN LINE                                ,3
       JK=3*K                                                      ,4
       NPTS=B(JK-1)                                                ,5
       LOCST=B(JK)                                                 ,6
       LOCEND=LOCST+3*NPTS-1
C      THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C        IN B(LOCST).
C      THE Z OF THE LAST POINT IS IN B(LOCEND).
C      FOR PAGE CONTROL WE WRITE A BLOCK CF POINTS FROM
C        B(LA) TO B(LB).                                           ,7
       LA=LOCST                                                    ,8
       NEWPCT=.TRUE.                                               ,9
       ISTA=1
C
C      WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN    ,10
  200  NELREM=NELREM-9
C      THIS ALLOWS FOR HEADING                                     ,11
       IF(NELREM-6)300,400,400
C      NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE        ,12
  300  WTITLE=.TRUE.                                               ,13
       NELREM=MAXEL-6                                              ,14
       GO TO 500                                                   ,15
  400  WTITLE=.FALSE.
C                                                                  ,16
  500  LB=MINO (LOCEND,LA+NELREM-1)                                ,17
       NL=(LB-LA+1)/3                                              ,18
       IF(.NOT. WTITLE) GO TO 600                                  ,19    ,20    ,21
       WRITE(LO,550)TITLE,DAYT                                     ,22    ,23    ,24
  550  FORMAT(1H1 12A6/54X 2A6)
  600  IF(NEWPCT) GO TO 700
C      WRITE HEADING                                               ,25    ,26    ,27
       WRITE(LO,650)K                                              ,28    ,29    ,30
```

```
          BODY1M                                                                   09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

  650 FORMAT(19H0BODY MERIDIAN LINE I4,14H   (CONTINUED))
      GO TO 800                                                         ,31
  700 WRITE(LO,750)K                                                    ,32   ,33   ,34
  750 FORMAT(19H0BODY MERIDIAN LINE I4)
      NUM=0                                                             ,35
  800 WRITE(LO,850)                                                     ,36   ,37
  850 FORMAT(14X1HX10X1HY10X1HZ12X3HRHO7X5HTHETA)
C
C     READY TO WRITE PCINTS. COMPUTING RHO-THETA AS WE GO
      DO 2000 L=1,NL                                                    ,38
      NUM=NUM+1                                                         ,39
      CALL TRAV(XYZ,B(LA),3)                                            ,40
      DY=XYZ(2)-AXIS(1)                                                 ,41
      DZ=XYZ(3)-AXIS(2)                                                 ,42
      RHO=SQRT(DY*DY+DZ*DZ)                                             ,43
      IF(RHO)1100,1100,1200                                             ,44
 1100 THETA=0.                                                          ,45
      GO TO 1300                                                        ,46
 1200 THETA = ATN1(ABS(DY),DZ)*DEGRAD
C                                                                       ,47
 1300 PNTR=BLANK
C     DOES THIS PT. LIE IN A DEFINING STATION PLANE                     ,48
      IF(ABS(XYZ-STA(ISTA))-1.E-4) 140C,1400,1500
C     YES.  PRINT *                                                     ,49
 1400 PNTR=STAR                                                         ,50
      ISTA=ISTA+1
C                                                                       ,51
 1500 WRITE(LO,1550)NUM,XYZ,PNTR,RHO,THETA                              ,52   ,53   ,54
 1550 FORMAT(I4,4X,3F11.4,A3,2F11.4)
      LA=LA+3                                                           ,55
 2000 CONTINUE
C                                                                       ,56   ,57
      NELREM=NELREM-3*AL                                                ,58
      IF(LB-LOCEND)300C,4000,4000                                       ,59
 3000 NEWPCT=.FALSE.                                                    ,60
      LA=LB+1                                                           ,61
      GO TO 200
C                                                                       ,62
 4000 CONTINUE
C                                                                       ,63   ,64
      RETURN                                                            ,65
      END                                                               ,66
```

```
C      USED BY BODY1 TO READ DEFINING SECTION DATA, PROCESS IT,
C        WRITE SECTION DATA, AND FORM MERIDIAN LINES.
C
       SUBROUTINE BODY1R(LI,LO,NPT,NSTA,BAXIS,TITLE,DAT,THETS,
      1 THETA,RHO,PT, BPL,STA,NU)
C
       DIMENSION BAXIS(2),TITLE(12),DAT(2),THETS(NPT),THETA(NPT)
      1,RHO(NPT), PT(3,NPT),BPL(3,NSTA,NPT),NU(3),SECAX(2),STA(NSTA)
C
       LOGICAL NEWPAG,NEWSTA
C
C      INPUTS ARE
C        LI = INPUT TAPE, LO = OUTPUT TAPE
C        NPT = NO. OF POINTS PER SECTION
C        NSTA = NO. OF STATIONS (DEFINING SECTIONS)
C        BAXIS = Y,Z OF BODY AXIS (AXIS PARALLEL TO X-AXIS)
C        TITLE, DAT = DATE
C        THETS = SET OF STANDARD THETA-VALUES (DEGREES)
C
C      SCRATCH ARRAYS ARE THETA, RHO AND PT
C
C      OUTPUTS ARE
C        BPL = XYZ COORD. OF POINTS ON MERIDIAN LINES
C        STA = ARRAY OF BODY STATIONS
C        NU(1) = 0 IF SUCCESS
C
       DATA LINPAG /45/
C      CHANGE LINPAG  IF 45 LINES/PAGE NOT OK
C      ASSUME 7 LINES HAVE BEEN WRITTEN ON PAGE (TITLE, ETC.)
C
       MAXLIN=LINPAG-2                                                   ,1
       LINREM=MAXLIN-5                                                   ,2
       NEWPAG=.FALSE.
C                                                                        ,3
       DO 2000 K=1,NSTA
C                                                                        ,4
       NEWSTA=.TRUE.
C      READ BODY SECTION DATA CARDS, COMPUTE PTS. ON SECTION             ,5
       CALL BODY1S(LI,BAXIS,NPT,THETS,SECAX,SCODE,THETA,RHO,PT,NU)
C      SECAX = Y,Z COORD. OF SECTION AXIS
C      NU=1 IF SCODE NOT BETWEEN 0. AND 6.
C      NU=2 IF SCODE = 3. (ELLIPSE) AND MAJOR OR MINOR AXIS N.G.         ,6
       IF(NU) 8000,100,8000                                              ,7
   100 STA(K)=PT                                                         ,8
       LINES=NPT                                                         ,9
       NUM=0
C                                                                        ,10
   200 LINREM=LINREM-5                                                   ,11
       IF(LINREM-6)300,400,400                                           ,12
   300 NEWPAG=.TRUE.                                                     ,13
       LINREM=MAXLIN                                                     ,14
       GO TO 500                                                         .15
   400 NEWPAG=.FALSE.
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C                                                                ,16
  500 NL=MINO(LINES,LINREM)                                      ,17
      IF(.NOT. NEWPAG)  GO TO 600                                ,18   ,19   ,20
      WRITE(LO,550)TITLE,DAT                                     ,21   ,22   ,23
  550 FORMAT(1H1 12A6/54X 2A6)
C
  600 IF(NEWSTA) GO TO 700                                       ,24   ,25   ,26
      WRITE(LO,610)PT(1,1),SCODE                                 ,27   ,28   ,29
  610 FORMAT(18HOBODY STATION  X = F10.4,13H  (CONTINUED)
     1 8X,8HSCODE  = F4.0)
      GO TO 800
C                                                                ,30
  700 WRITE(LO,710)PT(1,1),SCODE                                 ,31   ,32   ,33
  710 FORMAT(18HOBODY STATION  X = F10.4,21X,8HSCODE  = F4.0)
C
  800 WRITE(LO,810)SECAX                                         ,34   ,35   ,36
  810 FORMAT(41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = (
     1 F9.4,1H, F9.4, 1H)/13HOMERIDIAN NO. 4X,3HRHO 7X
     2 5HTHETA 18X 1HY 10X 1HZ)
C
      DO 900 L=1,NL                                              ,37
      NUM=NUM+1                                                  ,38
  900 WRITE(LO,910)NUM,RHO(NUM),THETA(NUM),(PT(I,NUM),I=2,3)     ,39   ,40   ,41   ,42   ,43   ,44
                                                                 ,45
  910 FORMAT(I9,F14.4,F11.4,F21.4,F11.4)
C
      LINES=LINES-NL                                             ,46
      LINREM=LINREM-NL                                           ,47
      IF(LINES)1100,1100,1000                                    ,48
 1000 NEWSTA=.FALSE.                                             ,49
      GO TO 200
C
C     STORE SECTION POINTS AS MERIDIAN LINES                     ,50
 1100 DO 1200 J=1,NPT                                            ,51
      DO 1200 I=1,3                                              ,52
 1200 BPL(I,K,J)=PT(I,J)
C                                                                ,53   ,5    ,55
 2000 CONTINUE
C                                                                ,56   ,57
 8000 RETURN                                                     ,58
      END                                                        ,59
```

```
C     USED BY BODY1R TO READ BODY SECTION CARDS, COMPUTE PTS. ON SECTION
C
      SUBROUTINE BODY1S(LI,BAXIS,NPT,THETS,SECAX,SCODE,THETA,RHO,PT,NU)
C
      DIMENSION BAXIS(2),THETS(NPT),SECAX(2),THETA(NPT),RHO(NPT),PT(3,NP
     1T) ,RAD(2),V(2),PTSEC(2),YZ(2)
C
C    INPUTS ARE
C     LI = INPUT TAPE NO.
C     BAXIS = Y,Z OF MAIN BODY AXIS
C     NPT = NO. OF PTS IN SECTION
C     THETS = STANDARD SET OF THETA-VALUES TO USE IF SCODE = 1,2 OR 3
C    OUTPUTS ARE
C     SECAX = Y,Z OF SECTION AXIS
C     SCODE = CODE FOR TYPE OF SECTION
C     THETA, RHO = POLAR COORD. OF PTS ON SECTION (ORIGIN AT SECAX)
C     PT = XYZ COORD. OF PTS ON SECTION
C     NU = 0 IF SUCCESS
C          1 IF SCODE .LT. 0 OR .GT. 6
C          2 IF SCODE = 3 (ELLIPSE) AND A SEMI-AXIS IS = 0.
C
      DATA DEGRAD/0206712273406/                                 180/PI
C
      READ(LI,100)STA,YZ,SCODE,RAD                                          ,1    ,2    ,3
 100  FORMAT(6F10.0)
      NU=1
C     CHECK FOR VALID SCODE                                                 ,4
      IF(SCODE .LT. 0. .OR. SCODE .GT. 6.) GO TO 8000                       ,5    ,6    ,7
 200  ICODE=SCODE                                                           ,8
      DO 210 I=1,2                                                          ,9
 210  SECAX(I)=BAXIS(I)+YZ(I)
C
C     IF ICODE .LE. 0, SECTION SAME AS PREVIOUS EX. FOR STATION             ,10   ,11
      IF(ICODE)5000,5000,300                                                ,12
 300  IF(.NOT. (ICODE .EQ. 1 .OR. ICODE .EQ. 5))GO TO 400
C     ICODE = 1 OR 5.   READ RHO                                            ,13   ,14   ,15
      READ(LI,100)(RHO(I),I=1,NPT)                                          ,16   ,17   ,18   ,19   ,20
 400  IF(ICODE .LT.4 .OR. ICODE .GT. 5)GO TO 500
C     ICODE = 4 OR 5.  READ THETA                                           ,21   ,22   ,23
      READ(LI,100)(THETA(I),I=1,NPT)                                        ,24   ,25   ,26   ,27   ,28
      GO TO 1000                                                            ,29
 500  IF(ICODE .GT. 3)GO TO 1000
C     ICODE = 1,2 OR 3.  STORE THETAS IN THETA                              ,30   ,31   ,32
      CALL TRAV(THETA,THETS, NPT)
C
C     THE TABLE BELOW SHOWS WHETHER RHO AND THETA HAVE BEEN FOUND
C     FOR EACH POINT ON THE SECTION
C
C     ICODE RHO THETA
C       0   YES  YES
C       1   YES  YES
C       2   NO   YES
C       3   NO   YES
```

```
C        4   NO    YES
C        5   YES   YES
C        6   NO    NO
C                                                                      ,33
 1000 GO TO (2000,1100,1200,1100,2000,4000),ICODE
C
C     CIRCLE (ICODE=2 OR 4)                                            ,34
 1100 DO 1110 I=1,NPT                                                  ,35
 1110 RHO(I)=RAD                                                       ,36    ,37
      GO TO 2000
C
C     ELLIPSE (ICODE=3)                                                ,38
 1200 DO 1400 I=1,NPT                                                  ,39
      THETR=THETA(I)/DEGRAD                                            ,40
      CALL ELLIPR(RAD,RAD(2),SIN(THETR),COS(THETR),RHO(I),X,Y,NU)      ,41
      MU=2                                                             ,42
      IF(NU)8000,1300,8000                                             ,43
 1300 PT(2,I)=X+SECAX(1)                                               ,44
      PT(3,I)=Y+SECAX(2)                                               ,45
 1400 CONTINUE                                                         ,46    ,47
      GO TO 5000
C
C     ICODE = 1,2,4 OR 5.  FIND Y,Z FOR EACH PT.                       ,48
 2000 DO 2010 I=1,NPT                                                  ,49
      THETR=THETA(I)/DEGRAD                                            ,50
      PT(2,I)=RHO(I)*SIN(THETR)+SECAX(1)                               ,51
 2010 PT(3,I)=RHO(I)*COS(THETR)+SECAX(2)                               ,52    ,53
      GO TO 5000
C
C     Y,Z GIVEN (ICODE=6).  READ Y,Z AND FIND RHO,THETA                ,54
 4000 READ(LI,100)((PT(I,J),I=2,3),J=1,NPT)                            ,55    ,56    ,57    ,58    ,59    ,60
                                                                       ,61    ,62
      DO 4400 I=1,NPT                                                  ,63
      PTSEC(1)=PT(2,I)                                                 ,64
      PTSEC(2)=PT(3,I)                                                 ,65
      RHO(I)=UVECN(SECAX,PTSEC,V,0,2)                                  ,66
      IF(RHO(I))4200,4200,4300                                         ,67
 4200 THETA(I)=THETS(I)                                                ,68
      GO TO 4400                                                       ,69
 4300 THETA(I)=ATN1(ABS(V),V(2))*DEGRAD                                ,70
 4400 CONTINUE
C
C     WE NOW HAVE RHO,THETA,Y,Z FOR EACH PT.  STORE STA AS X.          ,71    ,72
 5000 DO 5010 I=1,NPT                                                  ,73
 5010 PT(1,I)=STA                                                      ,74    ,75
      MU=0
C                                                                      ,76
 8000 NU=MU                                                            ,77
      RETURN                                                           ,78
      END                                                              ,79
```

```
       BSCALE                                                             07/31/65
    EXTERNAL FORMULA NUMBER   -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE BSCALE(B,NB,XEND,NU)
      DIMENSION B(1)
C
C     SCALE X-VALUES OF BODY MERIDIAN LINES SUCH THAT
C     EACH LINE STARTS AT X=0, AND ENDS WITH X=XEND.
C
C     X = (X-XF)*XEND/(XL-XF)
C     WHERE XF = X OF 1ST POINT OF A LINE, XL = X OF LAST POINT
C
      J=B(3)
C     1ST PT OF 1ST LINE STARTS IN B(J)
C                                                                          ,1
      DO 1000 K=1,NB
C                                                                          ,2
      XF=B(J)                                                              ,3
      NP=B(3*K-1)                                                          ,4
      L=J+3*(NP-1)                                                         ,5
      XL=B(L)                                                              ,6
      DEL=XL-XF                                                            ,7
      IF(DEL)200,100,200                                                   ,8
 100  NU=K                                                                 ,9
      GO TO 2000
C                                                                          ,10
 200  DO 300 I=1,NP                                                        ,11
      B(J)=((B(J)-XF)/DEL)*XEND                                            ,12
 300  J=J+3
C                                                                          ,13    ,14
 1000 CONTINUE                                                             ,15    ,16
      NU=0                                                                 ,17
 2000 RETURN                                                               ,18
      END                                                                  ,19
```

```
      SUBROUTINE BTHICK(NBODYS,NBODY,NROWB,XB,R,XC,ALPHA,THETA)
C     COMPUTES BODY THICKNESS SLOPES FOR GIVEN BODY RADII DISTRIBUTION
      DIMENSION XB(1),R(1),XC(1),ALPHA(1),THETA(1)

      NB=NBODYS-1
      NCOLB=NBODY/NROWB

      DO 100 I=1,NB
  100 ALPHA(I+1)=(R(I+1)-R(I))/(XB(I+1)-XB(I))
      ALPHA(1)=ALPHA(2)

      DO 250 J=1,NROWB
      DO 150 I=1,NB
      IF (XB(I)-XC(J)) 150,200,200
  150 CONTINUE

      I=NBODYS
  200 THETA(J)=ALPHA(I-1)+(XC(J)-XB(I-1))/(XB(I)-XB(I-1))*(ALPHA(I)-
     1ALPHA(I-1))
  250 CONTINUE

      DO 400 J=1,NROWB
      DO 400 I=2,NCOLB
      IJ=(I-1)*NROWB+J
      THETA(IJ)=THETA(J)
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CAMBER(NCOLW,NROWW,CHORD,ALPHAW,CSHAPE,CHORDL)
      DIMENSION CHORD(1),ALPHAW(1),CSHAPE(1),CHORDL(1)
C     COMPUTES SURFACE SHAPE, GIVEN SURFACE SLOPES
      DO 395 J=1,NCOLW
      CHORDL(J)=0.
      DO 392 I=1,NROWW
      K=(J-1)*NROWW+I
  392 CHORDL(J)=CHORDL(J)+CHORD(K)
      DO 395 I=1,NROWW
      SUM=0.
      IW=I
      DO 394 IJ=1,IW
      K=(J-1)*NROWW+IJ
  394 SUM=SUM+ALPHAW(K)*CHORD(K)/CHORDL(J)
      CSHAPE(K)=SUM
  395 CONTINUE
      RETURN
      END
```

```
       CAMBW                                                          08/16/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CAMBW (NW,NTAPEX,A,CLW,ALPHAW)
      DIMENSION A(1),ALPHAW(1),CLW(1)

C     FOR WING ONLY CASE, COMPUTES NORMAL VELOCITY COMPONENTS ON WING

      DO 100 J=1,NW                                              ,1
      ALPHAW(J)=0.0                                              ,2
  100 CONTINUE                                                   ,3    ,4
      DO 200 J=1,NW                                              ,5
      READ (NTAPEX) (A(I),I=1,NW)                                ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NW                                              ,11
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)                            ,12
  200 CONTINUE
                                                                 ,13   ,14   ,15
      RETURN                                                     ,16
      END                                                        ,17
```

```
      CAMBWB                                                                        08/16/65
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CAMBWB (NB,NW,NTAPEX,NTAPEY,A,CLW,ALPHAB,ALPHAW)

C     FOR WING-BODY CASES, COMPUTES NORMAL VELOCITY COMPONENTS ON WING

      DIMENSION A(1),ALPHAW(1),ALPHAB(1),CLW(1)

      DO 100 J=1,NW                                              ,1
      ALPHAW(J)=0.                                               ,2
  100 CONTINUE                                                   ,3    ,4

      DO 200 J=1,NW                                              ,5
      READ (NTAPEX) (A(I),I=1,NW)                                ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NW                                              ,11
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)                            ,12
  200 CONTINUE                                                   ,13   ,14   ,15

      DO 300 J=1,NB                                              ,16
      READ (NTAPEY) (A(I),I=1,NW)                                ,17   ,18   ,19   ,20   ,21
      DO 300 I=1,NW                                              ,22
      ALPHAW(I)=ALPHAW(I)+A(I)*ALPHAB(J)                         ,23
  300 CONTINUE                                                   ,24   ,25   ,26

      RETURN                                                     ,27
      END                                                        ,28
```

```
         CEGAR                                                        07/31/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CEGAR(N,P,AREA,CGRAV)
C
C          FINDS AREA AND CENTROID OF AREA OF A POLYGON
C
      DIMENSION P(2,1),CGRAV(2)
      DOUBLE PRECISION A,CG(2)
C
      A=0.D0                                                           ,1
      CG(1)=A                                                          ,2
      CG(2)=A                                                          ,3
      P(1,N+1)=P(1,1)                                                  ,4
      P(2,N+1)=P(2,1)
C                                                                      ,5
      DO 100J=1,2                                                      ,6
      K=(J+1)/J                                                        ,7
      DO 80I=1,N                                                       ,8
   80 CG(J)=CG(J)+(P(J,I+1)**2+P(J,I)**2+P(J,I+1)*P(J,I))*(P(K,I+1)-P(K,
     1I))                                                              ,9     ,10
  100 CONTINUE
C                                                                      ,11    ,12
      DO 120L=1,N                                                      ,13
  120 A=A+(P(2,L+1)+P(2,L))*(P(1,L+1)-P(1,L))
C                                                                      ,14    ,15
      IF (A) 160,140,160                                               ,16
  140 CG(1)=0.                                                         ,17
      CG(2)=0.                                                         ,18
      GO TO 180                                                        ,19
  160 CG(1)=CG(1)/(-A*3.)                                              ,20
      CG(2)=CG(2)/(A*3.)                                               ,21
      A=ABS (A)/2.                                                     ,22
  180 CONTINUE                                                         ,23
      AREA=A                                                           ,24
      CGRAV(1)=CG(1)                                                   ,25
      CGRAV(2)=CG(2)                                                   ,26
      RETURN                                                           ,27
       END                                                             ,28
```

```
      CENTRD                                                    08/09/65
      EXTERNAL FORMULA NUMBER  -  SOURCE STATEMENT  -  INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CENTRC (X,Y,Z,R,XC,YC,ZC,NC,NR)
      DIMENSION X(16,16),Y(16,16),Z(16,16),R(15,15),XC(15,15),
     1YC(15,15),ZC(15,15)

C     SUBROUTINE TO CALCULATE BODY AND WING PANEL
C     CENTROID COORDINATES

C     DEFINE FUNCTION TO BE USED TO CALCULATE
C     PANEL CENTROID COORDINATE
      CENT(A.,A2,A3,P1,P2)=A1+P1*(A2+P2*(A3-A2)-A1)

C     DEFINE CONSTANTS
      P1=2./3.                                                  ,1
      P2=.5

C     CALCULATE PANEL CENTROID COORDINATES                      ,2
 50   DO 60 I=1,NC                                              ,3
      DO 60 J=1,NR                                              ,4
      XC1=CENT(X(I,J+1),X(I,J),X(I+1,J+1),P1,P2)                ,5
      YC1=CENT(Y(I,J+1),Y(I,J),Y(I+1,J+1),P1,P2)                ,6
      ZC1=CENT(Z(I,J+1),Z(I,J),Z(I+1,J+1),P1,P2)                ,7
      XC2=CENT(X(I+1,J),X(I,J),X(I+1,J+1),P1,P2)                ,8
      YC2=CENT(Y(I+1,J),Y(I,J),Y(I+1,J+1),P1,P2)                ,9
      ZC2=CENT(Z(I+1,J),Z(I,J),Z(I+1,J+1),P1,P2)                ,10
      XC(I,J)=XC2+R(I,J)*(XC1-XC2)                              ,11
      YC(I,J)=YC2+R(I,J)*(YC1-YC2)                              ,12
 60   ZC(I,J)=ZC2+R(I,J)*(ZC1-ZC2)                              ,13   ,14   ,15
 100  RETURN                                                    ,16
      END                                                       ,17
```

```
                 CHECK                                                                  08/09/65
              EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CHECK (X,N,K,L,N6)
      DIMENSION X(1)
C     SUBROUTINE TO CHECK IF SEQUENCE IS
C     MONOTONIC INCREASING OR DECREASING
   10 IF (K .LT. -2 .OR. K .EQ. 0 .OR. K .GT. 2) GO TO 80          ,1   ,2   ,3
      K1=(K+IABS(K))/(2*K)                                         ,4
      K2=K1-1                                                      ,5
      N1=N-1                                                       ,6
   30 DO 60 I=1,N1                                                 ,7   ,8
      L1=K1*(I+1)+K2*(N-I)                                         ,9
      L2=L1-K1+K2                                                  ,10
      IF (IABS(K)-1) 80,40,50                                      ,11
   40 IF (X(L1)-X(L2)) 90,60,60                                    ,12
   50 IF (X(L1)-X(L2)) 90,90,60                                    ,13
   60 CONTINUE                                                     ,14  ,15
      L=0                                                          ,16
      GO TO 100                                                    ,17
   80 WRITE (N6,900)                                               ,18  ,19
      WRITE (N6,910) K                                             ,20  ,21  ,22
      L=1                                                          ,23
      GO TO 100                                                    ,24
   90 WRITE (N6,900)                                               ,25  ,26
      WRITE (N6,920) (X(I),I=1,N)                                  ,27  ,28  ,29  ,30  ,31
      L=1                                                          ,32
      GO TO 100                                                    ,33
  100 RETURN                                                       ,34
  900 FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHECK)
  910 FORMAT (1H ,9X,28HINVALID VALUE FOR K      K = ,I3)
  920 FORMAT (1H ,9X,25HSEQUENCE IS NOT MONOTONIC//
     1(F20.5))
      END                                                          ,35
```

```
                 CHORDW                                                        09/14/65
          EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

       SUBROUTINE CHORDW
       COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
      1NWRITE,NBODY,NWING,XMACH,SYM,KACE
       COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
      1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
      2,NPLANE,NPLN1
       COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
      1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
      2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
      3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
      4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
      5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
       DIMENSION C(4),P(3,50),Q(3,5),LOC(5)

C      SUBROUTINE TO CALCULATE WING PANEL STREAMWISE
C      CHORD LENGTHS

C      DEFINE CONSTANTS
       E=.000001                                                        ,1
       F=.001                                                           ,2
       M=3                                                              ,3
       MAX=2                                                            ,4
       N=4                                                              ,5
       C(1)=0.                                                          ,6
       C(2)=-1.                                                         ,7
       C(3)=0.

C      CALCULATE PANEL CHORD LENGTHS                                    ,8
 50    DO 70 I=1,NPLN1                                                  ,9
       DO 60 J=1,NPER1

C      SETUP ARRAY                                                      ,10
       P(1,1)=XCOR(I,J)                                                 ,11
       P(2,1)=YCOR(I,J)                                                 ,12
       P(3,1)=ZCOR(I,J)                                                 ,13
       P(1,2)=XCOR(I+1,J)                                               ,14
       P(2,2)=YCOR(I+1,J)                                               ,15
       P(3,2)=ZCOR(I+1,J)                                               ,16
       P(1,3)=XCOR(I+1,J+1)                                             ,17
       P(2,3)=YCOR(I+1,J+1)                                             ,18
       P(3,3)=ZCOR(I+1,J+1)                                             ,19
       P(1,4)=XCOR(I,J+1)                                               ,20
       P(2,4)=YCOR(I,J+1)                                               ,21
       P(3,4)=ZCOR(I,J+1)                                               ,22
       C(4)=YCON(I,J)

C      ATTEMPT INTERSECTION                                             ,23
       CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                            ,24
       IF (NINT-2) 80,60,90                                             ,25
 60    CHORD(I,J)=ABS(Q(1,1)-Q(1,2))                                    ,26    ,27
 70    CONTINUE                                                         ,28    ,29
       GO TO 100
```

```
      CHORDW                                                        09/14/65
   EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C     ERROR MESSAGES                                                    ,30
 80   WRITE (NWRITE,900)                                                ,31    ,32
      GO TO 95                                                          ,33
 90   WRITE (NWRITE,910)                                                ,34    ,35
      GO TO 95                                                          ,36
 95   KODEW=1                                                           ,37
      GO TO 100                                                         ,38
100   RETURN                                                            ,39
900   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHORD/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     133HPANEL CHORD LENGTH NOT CALCULATED/10X,
     237HFEWER THAN TWO INTERSECTIONS COMPUTED)
910   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CHORD/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     133HPANEL CHORD LENGTH NOT CALCULATED/10X,
     236HMORE THAN TWO INTERSECTIONS COMPUTED)
      END                                                               ,40
```

```
     CLEAR                                                      10/04/65
    7094 RELMOD ASSEMBLY.

          $IBLDR CLEAR                                                                  CLEA0000


    ASSEMBLED TEXT.
                                                                                        CLEA0001
          $TEXT  CLEAR
                                    *CLEAR      MICHAEL SYNGE          6-9247      ASSEMBLED  21 SEP 64
                                    *  TO CLEAR ANY NUMBER OF STORAGE BUFFERS
                                    *
                                    *    CALLING SEQUENCE -
                                    *
                                    *                CALL CLEAR (BUFA,BUFB,.......)
                                    *
                                          ENTRY    CLEAR
                                    *
BINARY CARD ID. CLEA0002
        00000  0634 00 4 00004  10001   CLEAR SXA    EXIT,4
        00001  0500 00 4 00003  10000         CLA    3,4
        00002  4320 00 0 00021  10001         ANA    =O777777700000
        00003  0100 00 0 00006  10001         TZE    NEXT
        00004  0774 00 4 00000  10000   EXIT  AXT    0,4
        00005  0020 00 4 00001  10000         TRA    1,4
                                        *
        00006  0500 60 4 00003  10000   NEXT  CLA*   3,4
        00007  0621 00 0 00020  10001         STA    LENG
        00010  0500 00 4 00003  10000         CLA    3,4
        00011  0621 00 0 00016  10001         STA    BUF
        00012  0634 00 4 00014  10001         SXA    XR,4
        00013  0C74 00 4 03000  10011         TSX    $INIBFR,4
        00014  0774 00 4 00000  10000   XR    AXT    **,4
        00015  1 77777 4 02001  10011         TXI    CLEAR+1,4,-1
        00016  0000 00 0 00000  10000   BUF   HTR    **
        00017  0000 00 0 00020  10001         HTR    LENG
                                        *
        00020  0 00000 0 00000  10000   LENG
        00021  777777700000     10000         *LORG
                          00000 01111         END


    CONTROL DICTIONARY

          $CDICT CLEAR                                                                  CLEA0003


BINARY CARD ID. CLEA0004
             000022000000              PREFACE     START=0,LENGTH=18,TYPE=7094,CMPLX=5
             000004000005
             234325215160       CLEAR  DECK        LOC=0,LENGTH=18
             000022000000
             234325215160       CLEAR  REAL        LOC=0,LENGTH=0
             000000000000
             314531222651       INIBFR VIRTUAL     SECT. 3
```

```
$UKEND CLEAR                                                        CLEA0005

NO MESSAGES FOR THIS ASSEMBLY

           SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL    VALUE    REFERENCES

         BUF       00016    11
         CLEAR     00000    15
         EXIT      00004    0
         LENG      00020    7,17
         NEXT      00006    3
  LCTR    BLCTR
  QUAL    UNQS              13
  LCTR   //
         XR        00014    12

REFERENCES TO VIRTUAL SYMBOLS.

         INIBFR        3    13
```

```
      CLOK                                                              08/16/65
   EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CLOK(TIME,DATE)
C     DUMMY ROUTINE TO RETURN TIME, DATE AS BLANKS.
C     IT MAY BE REPLACED BY A ROUTINE WHICH RETURNS TWO
C     HOLLERITH WORDS CONTAINING THE TIME AND DATE.
      DATA T/1H /
      TIME = T                                                          .1
      DATE = T                                                          .2
      RETURN                                                            .3
      END                                                               .4
```

```
       CNTRLB                                                          08/16/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CNTRLB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KORTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(15),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),P(3,10),Q(3,5),LOC(5)
C     SUBROUTINE TO CALCULATE BODY PANEL
C     CONTROL POINT COORDINATES AND PANEL CHORD LENGTHS
C     DEFINE FUNCTIONS TO BE USED IN CALCULATIONS
C     XLENG CALCULATE LENGTH OF STREAMWISE CHORD
C     THROUGH PANEL CENTROID
      XLENG(X1,X2,Y1,Y2,Z1,Z2)=SQRT((X2-X1)**2+(Y2-Y1)**2+
     1(Z2-Z1)**2)
C     DEFINE CONSTANTS
 10   M=3                                                              ,1
      MAX=2                                                            ,2
      N4=4                                                             ,3
      N5=5                                                             ,4
      E=0.001                                                           5
      F=0.000001
C     CALCULATE PANEL CONTROL POINT COORDINATES
C     AND PANEL CHORD LENGTHS                                          ,6
 50   DO 90 I=1,NPER1                                                  ,7
      DO 80 J=1,NPLN1                                                  ,8
      IF (KODEC) 150,53,57                                             ,9
 53   YTEMP=YCEN(I,J)                                                  ,10
      ZTEMP=ZCEN(I,J)                                                  ,11
      GO TO 60                                                         ,12
 57   YTEMP=YCOR(I,J)+YPER*(YCOR(I+1,J+1)-YCOR(I,J))                    13
      ZTEMP=ZCOR(I,J)+YPER*(ZCOR(I+1,J+1)-ZCOR(I,J))
C     SETUP ARRAY                                                      ,14
 60   P(1,1)=XCOR(I,J+1)                                               ,15
      P(2,1)=YCOR(I,J+1)                                               ,16
      P(3,1)=ZCOR(I,J+1)                                               ,17
      P(1,2)=XCOR(I,J)                                                 ,18
      P(2,2)=YCOR(I,J)                                                 ,19
      P(3,2)=ZCOR(I,J)                                                 ,20
      P(1,3)=XCOR(I+1,J)                                               ,21
      P(2,3)=YCOR(I+1,J)                                               ,22
      P(3,3)=ZCOR(I+1,J)                                                23
      P(1,4)=XCOR(I+1,J+1)                                             ,24
```

```
      P(2,4)=YCOR(I+1,J+1)                                        ,25
      P(3,4)=ZCOR(I+1,J+1)                                        ,26
      P(1,5)=XCOR(I,J+1)                                           27
      P(2,5)=YCOR(I,J+1)                                          ,28
      P(3,5)=ZCOR(I,J+1)                                          ,29
      IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 63,63,67              ,30
 63   C(1)=0.                                                     ,31
      C(2)=-1.                                                    ,32
      C(3)=0.                                                     ,33
      C(4)=YTEMP                                                  ,34
      GO TO 70                                                    ,35
 67   C(1)=0.                                                     ,36
      C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1)) ,37
      C(3)=-1.                                                    ,38
      C(4)=ZTEMP-C(2)*YTEMP

C     ATTEMPT INTERSECTION                                        ,39
 70   CALL POLXN (C,P,M,N5,F,MAX,Q,LOC,NINT)                      ,40
      IF (NINT-2) 143,73,145                                      ,41
 73   CHORD(I,J)=XLENG(Q(1,1),Q(1,2),Q(2,1),Q(2,2),Q(3,1),Q(3,2)) ,42
      IF (Q(1,1)-Q(1,2)) 75,150,77                                ,43
 75   XCON(I,J)=Q(1,1)+XPER*(Q(1,2)-Q(1,1))                       ,44
      YCON(I,J)=Q(2,1)+XPER*(Q(2,2)-Q(2,1))                       ,45
      ZCON(I,J)=Q(3,1)+XPER*(Q(3,2)-Q(3,1))                       ,46
      GO TO 80                                                    ,47
 77   XCON(I,J)=Q(1,2)+XPER*(Q(1,1)-Q(1,2))                       ,48
      YCON(I,J)=Q(2,2)+XPER*(Q(2,1)-Q(2,2))                       ,49
      ZCON(I,J)=Q(3,2)+XPER*(Q(3,1)-Q(3,2))                       ,50
 80   CONTINUE                                                    ,51   ,52
 90   CONTINUE                                                    ,53   ,54
      GO TO 150

C     ERROR MESSAGES                                              ,55
 143  WRITE (NWRITE,900)                                          ,56   ,57
      GO TO 147                                                    58
 145  WRITE (NWRITE,910)                                          ,59   ,60
      GO TO 147                                                   ,61
 147  KODEB=1                                                     ,62
      GO TO 150                                                    63
 150  RETURN                                                      ,64
 900  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLB/10X,
     134HPANEL CONTROL POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     237HFEWER THAN TWO INTERSECTIONS COMPUTED)
 910  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLB/10X,
     134HPANEL CONTROL POINT NOT CALCULATED/10X,
     124HSUBROUTINE POLXN FAILURE/10X,
     236HMORE THAN TWO INTERSECTIONS COMPUTED)
       END                                                        ,65
```

```
          CNTRLW                                                    09/14/65
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

         SUBROUTINE CNTRLW
         COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
        1NWRITE,NBODY,NWING,XMACH,SYM,KACE
         COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
        1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
        2,NPLANE,NPLN1
         COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
        1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
        2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
        3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
        4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
        5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)

C        SUBROUTINE TO CALCULATE WING PANEL CONTROL POINT
C        COORDINATES
 10      IF (KSTART-1) 15C,60,20

C        CALCULATE PANEL CONTROL POINT COORDINATES ON
C        INBOARD COLUMN IF BODY AND WING INTERSECT IN
C        NON-STREAMWISE LINE                                                    ,1
 20      DO 55 J=1,NPER1                                                        ,2
         I=1                                                                    ,3
         IF (KODEC) 150,23,27                                                   ,4
 23      YCON(I,J)=YCEN(I,J)                                                    ,5
         GO TO 30                                                               ,6
 27      YCON(I,J)=YCOR(I+1,J)+YPER*(AMAX1(YCOR(I,J),YCOR(I,J+1))-
        1YCOR(I+1,J))                                                           ,7
 30      IF (YCOR(I,J)-YCON(I,J)) 33,33,35                                      ,8
 33      YPERL=(YCON(I,J)-YCOR(I+1,J))/(YCOR(I,J)-YCOR(I+1,J))                  ,9
         XLEAD=XCOR(I+1,J)+YPERL*(XCOR(I,J)-XCOR(I+1,J))                        ,10
         ZLEAD=ZCOR(I+1,J)+YPERL*(ZCOR(I,J)-ZCOR(I+1,J))                        ,11
         GO TO 40                                                               ,12
 35      IF (YCOR(I,J+1)-YCON(I,J)) 37,37,140                                   ,13
 37      YPERL=(YCON(I,J)-YCOR(I,J))/(YCOR(I,J+1)-YCOR(I,J))                    ,14
         XLEAD=XCOR(I,J)+YPERL*(XCOR(I,J+1)-XCOR(I,J))                          ,15
         ZLEAD=ZCOR(I,J)+YPERL*(ZCOR(I,J+1)-ZCOR(I,J))                          ,16
 40      IF (YCOR(I,J+1)-YCON(I,J)) 43,43,45                                    ,17
 43      YPERT=(YCON(I,J)-YCOR(I+1,J+1))/(YCOR(I,J+1)-YCOR(I+1,J+1))            ,18
         XTRAIL=XCOR(I+1,J+1)+YPERT*(XCOR(I,J+1)-XCOR(I+1,J+1))                 ,19
         ZTRAIL=ZCOR(I+1,J+1)+YPERT*(ZCOR(I,J+1)-ZCOR(I+1,J+1))                 ,20
         GO TO 50                                                               ,21
 45      IF (YCOR(I,J)-YCON(I,J)) 47,47,140                                     ,22
 47      YPERT=(YCON(I,J)-YCOR(I,J+1))/(YCOR(I,J)-YCOR(I,J+1))                  ,23
         XTRAIL=XCOR(I,J+1)+YPERT*(XCOR(I,J)-XCOR(I,J+1))                       ,24
         ZTRAIL=ZCOR(I,J+1)+YPERT*(ZCOR(I,J)-ZCOR(I,J+1))                       ,25
 50      XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                                    ,26
         ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                                    ,27
    55   CONTINUE                                                               ,28    ,29
 60      IF (KEND-1) 150,70,70

C        CALCULATE PANEL CONTROL POINT COORDINATES FOR
C        MAIN WING REGION AND WING-TIP REGION                                   ,30
 70      DO 90 I=KSTART,KEND                                                    ,31
```

```
        CNTRLW                                                                09/14/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

       DO 90 J=1,NPER1                                                    ,32
       IF (KODEC) 150,73,77                                               ,33
 73    YCON(I,J)=YCEN(I,J)                                                ,34
       YPER=(YCON(I,J)-YCOR(I,J))/(YCOR(I+1,J)-YCOR(I,J))                 ,35
       GO TO 80                                                           ,36
 77    YCON(I,J)=YCOR(I,J)+YPER*(YCOR(I+1,J)-YCOR(I,J))                   ,37
 80    XLEAD=XCOR(I,J)+YPER*(XCOR(I+1,J)-XCOR(I,J))                       ,38
       XTRAIL=XCOR(I,J+1)+YPER*(XCOR(I+1,J+1)-XCOR(I,J+1))                ,39
       ZLEAD=ZCOR(I,J)+YPER*(ZCOR(I+1,J)-ZCOR(I,J))                       ,40
       ZTRAIL=ZCOR(I,J+1)+YPER*(ZCOR(I+1,J+1)-ZCOR(I,J+1))                ,41
       XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                                ,42
 90    ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                                ,43   ,44   ,45
       IF (KEND-NPLN1) 100,150,150                                        ,46
 100   I=NPLN1                                                            ,47
       DO 135 J=1,NPER1                                                   ,48
       IF (KODEC) 150,1C3,107                                             ,49
 103   YCON(I,J)=YCEN(I,J)                                                ,50
       GO TO 110                                                          ,51
 107   YCON(I,J)=YCOR(I,J)+YPER*(AMIN1(YCOR(I+1,J),
      1YCOR(I+1,J+1))-YCOR(I,J))                                          ,52
 110   IF (YCON(I,J)-YCOR(I+1,J)) 113,113,115                             ,53
 113   YPERL=(YCON(I,J)-YCOR(I,J))/(YCOR(I+1,J)-YCOR(I,J))                ,54
       XLEAD=XCOR(I,J)+YPERL*(XCOR(I+1,J)-XCOR(I,J))                      ,55
       ZLEAD=ZCOR(I,J)+YPERL*(ZCOR(I+1,J)-ZCOR(I,J))                      ,56
       GO TO 120                                                          ,57
 115   IF (YCON(I,J)-YCOR(I+1,J+1)) 117,117,140                           ,58
 117   YPERL=(YCON(I,J)-YCOR(I+1,J))/(YCOR(I+1,J+1)-YCOR(I+1,J))          ,59
       XLEAD=XCOR(I+1,J)+YPERL*(XCOR(I+1,J+1)-XCOR(I+1,J))                ,60
       ZLEAD=ZCOR(I+1,J)+YPERL*(ZCOR(I+1,J+1)-ZCOR(I+1,J))                ,61
 120   IF (YCON(I,J)-YCOR(I+1,J+1)) 123,123,125                           ,62
 123   YPERT=(YCON(I,J)-YCOR(I,J+1))/(YCOR(I+1,J+1)-YCOR(I,J+1))          ,63
       XTRAIL=XCOR(I,J+1)+YPERT*(XCOR(I+1,J+1)-XCOR(I,J+1))               ,64
       ZTRAIL=ZCOR(I,J+1)+YPERT*(ZCOR(I+1,J+1)-ZCOR(I,J+1))               ,65
       GO TO 130                                                          ,66
 125   IF (YCON(I,J)-YCOR(I+1,J)) 127,127,140                             ,67
 127   YPERT=(YCON(I,J)-YCOR(I+1,J+1))/(YCOR(I+1,J)-YCOR(I+1,J+1))        ,68
       XTRAIL=XCOR(I+1,J+1)+YPERT*(XCOR(I+1,J)-XCOR(I+1,J+1))             ,69
       ZTRAIL=ZCOR(I+1,J+1)+YPERT*(ZCOR(I+1,J)-ZCOR(I+1,J+1))             ,70
 130   XCON(I,J)=XLEAD+XPER*(XTRAIL-XLEAD)                                ,71
 135   ZCON(I,J)=ZLEAD+XPER*(ZTRAIL-ZLEAD)                                ,72   ,73
       GO TO 150                                                          ,74
 140   WRITE (NWRITE,90C)                                                 ,75   ,76
       GO TO 150                                                          ,77
 150   RETURN                                                             ,78
 900   FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE CNTRLW/10X,
      157HPANEL CONTROL POINT Y-COORDINATE NOT CORRECTLY CALCULATED)
       END                                                                ,79
```

```
      COEFS      - EFN   SOURCE STATEMENT  -  IFN(S)  -                05/22/67

      SUBROUTINE COEFS(XB,R,FD,THETAB,CPBB,NFT,NLT,NTHETA,XP,ZP,C,IRROR)

C     SUBROUTINE TO COMPUTE CX (ENTRY POINT SCX), CN
C     (ENTRY POINT SCN) AND CM (ENTRY POINT SCM)

      DIMENSION XB(1),R(1),FD(1),THETAB(1),CPBB(10,1)
      DIMENSION Y(55),YY(10),S(55),T(10)

      ENTRY SCX(XB,R,FD,THETAB,CPBB,NFT,NLT,NTHETA,XP,ZP,C,IRROR)
      DO 100 I=  NFT,NLT
  100 S(I)=FD(I)
      DO 200 I=1,NTHETA
  200 T(I)=0.
      CC= 1.
      GO TO 1000

      ENTRY SCN(XB,R,FD,THETAB,CPBB,NFT,NLT,NTHETA,XP,ZP,C,IRROR)
      DO 400 I= NFT,NLT
  400 S(I)=1.
      DO 500 I=1,NTHETA
  500 T(I)= THETAB(I)
      CC=-1.
      GO TO 1000

      ENTRY SCM(XB,R,FD,THETAB,CPBB,NFT,NLT,NTHETA,XP,ZP,C,IRROR)
      DO 700 I=  NFT,NLT
  700 S(I)=XB(I)-XP
      DO 800 I=1,NTHETA
  800 T(I)= THETAB(I)
      CC=-1.

 1000 DO 1100 I= NFT,NLT
      DO 1050 J=1,NTHETA
 1050 YY(J)=CC*CPBB(J,I)*COS(T(J))
      CALL SIMUN3(THETAB,YY,NTHETA,A,IRR)                                 60
      IF (IRR .NE. 0) GO TO 2000                                          64
 1100 Y(I)=R(I)*S(I)*A
      NCT=NLT-NFT+1
      CALL SIMUN3(XB,Y,NCT    ,A,IRR)                                     76
      IF (IRR .NE. 0) GO TO 2000
      C=A
      IRROR=0
 1500 RETURN

 2000 IRROR=IRR
      GO TO 1500
      END
```

```
      SUBROUTINE COMCU(DA,DB,S,X,Y,L,M,N,NDA,NDB)                     COMC0060
C                                                                     COMC0070
C        (FOR CDC 6600, 7/10/66)                                      COMC0045
C                                                                     COMC0050
      DIMENSION C(400), D(400), E(400), S(1), X(1), Y(1)              COMC0090
C                                                                     COMC0100
      K=N-1                                                           COMC0110
      KUE=0                                                           COMC0120
      IF (N-2) 5,10,30                                                COMC0130
    5 M=-1                                                            COMC0140
      GO TO 260                                                       COMC0150
   10 IF (NDA-1) 25,15,25                                             COMC0160
   15 IF (NDB-1) 25,20,25                                             COMC0170
   20 S(1)=DA                                                         COMC0180
      S(2)=DB                                                         COMC0190
      M=0                                                             COMC0200
      GO TO 260                                                       COMC0210
   25 KUE=1                                                           COMC0220
   30 M=0
   40 E(1)=0.0
      C(N)=0.0                                                        COMC0270
      IF (NDA-1) 50,50,60                                             COMC0280
   50 D=1.0                                                           COMC0290
      C=0.0                                                           COMC0300
      S=DA                                                            COMC0310
      GO TO 70                                                        COMC0320
   60 D=4.0                                                           COMC0330
      C=2.0                                                           COMC0340
      S=6.0*(Y(2)-Y(1))/(X(2)-X(1))-DA*(X(2)-X(1))                    COMC0350
   70 IF (KUE) 85,75,85                                               COMC0360
   75 DO 80 I=2,K                                                     COMC0370
      U =     X(I)-X(I-1)
      V =     X(I+1)-X(I)
      C(I)=U                                                          COMC0420
      D(I)=2.0* (U+V)
      E(I)=V                                                          COMC0400
   80 S(I)=3.0/(U*V) * ( U*U*(Y(I+1)-Y(I)) + V*V*(Y(I)-Y(I-1)) )
   85 IF (NDB-1) 90,90,100                                            COMC0450
   90 E(N)=0.0                                                        COMC0460
      D(N)=1.0                                                        COMC0470
      S(N)=DB                                                         COMC0480
      GO TO 110                                                       COMC0490
  100 E(N)=2.0                                                        COMC0500
      D(N)=4.0                                                        COMC0510
      S(N)=6.0*(Y(N)-Y(N-1))/(X(N)-X(N-1))+DB*(X(N)-X(N-1))           COMC0520
  110 C=C/D                                                           COMC0530
      S=S/D                                                           COMC0540
      DO 120 I=2,N                                                    COMC0550
      F = D(I) - C(I-1)*E(I)
      C(I)=C(I)/F                                                     COMC0570
  120 S(I) = ( S(I)-S(I-1)*E(I) )/F
      DO 130 J=1,K                                                    COMC0590
      I=N-J                                                           COMC0600
  130 S(I) = S(I) - S(I+1)*C(I)
  260 RETURN                                                          COMC0650
      END                                                             COMC0660
```

```
    COMP         -  EFN   SOURCE STATEMENT  -  IFN(S)  -                05/31/67

   SUBROUTINE COMP(GP,VLS,IC)
   DIMENSION GP(3,1),VLS(4,1)
   DATA ISTR/0/
   DO 10 I=1,IC
10 CALL LACKEY(ISTR,          GP(1,I),GP(2,I),GP(3,I),VLS(1,I),VLS(2,
  $I),VLS(3,I),VLS(4,I))                                                    13
   RETURN
   END
```

```
      SUBROUTINE CP(NP,XMACH,CPCALC,U,V,W,CPP)

C     SUBROUTINE TO COMPUTE SECOND ORDER OR EXACT
C     ISENTROPIC COEFFICIENT OF PRESSURE

      DIMENSION U(1),V(1),W(1),CPP(1)

      XM2=XMACH*XMACH

      IF (CPCALC-1.) 200,200,300

C     SECOND ORDER CP CALCULATIONS
  200 CONTINUE
      DO 250 J=1,NP
      CPP(J)=-2.*U(J)+XM2*U(J)*U(J)
  250 CONTINUE
      GO TO 999

C     EXACT ISENTROPIC CP CALCULATIONS
  300 CONTINUE
      DO 350 J=1,NP
      UVW2=U(J)**2+V(J)**2+W(J)**2
      CPP(J)=1.42857/XM2*((1.-.2*XM2*(2.*U(J)+UVW2))**3.5 -1.)          26
  350 CONTINUE

  999 RETURN
      END
```

```
      SUBROUTINE CRNRB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(4),D(4),P(3,50),R(3,50),Q(3,5),LOC(3)
      DIMENSION ISAVE(21)

C     SUBROUTINE TO CALCULATE BODY PANEL CORNER
C     POINT COORDINATES

C     DEFINE CONSTANTS
      EPS=0.04                                                      ,1
      E=0.001                                                       ,2
      F=0.0001                                                      ,3
      M=3                                                           ,4
      MAX=1                                                         ,5
      N=2

C     CALCULATE X-, Y- AND Z-COORDINATES OF PANEL
C     CORNER POINTS     PREVIOUSLY DETERMINED BODY
C     PERCENT LINES ARE INTERSECTED WITH SERIES
C     OF PLANES PARALLEL TO X-Z PLANE                               ,6
      IF (KOPTB) 300,3,13                                           ,7
 3    DO 10 I=1,NPER                                                ,8    ,9
      NPOINT=NPTS(I)                                                ,10
      DO 5 J=1,NPOINT                                               ,11   ,12
      P(1,J)=X(I,J)                                                 ,13
      P(2,J)=Y(I,J)                                                 ,14
 5    P(3,J)=Z(I,J)                                                 ,15   ,16
      DO 10 J=1,NPLANE                                              ,17   ,18
      C(1)=1.                                                       ,19
      C(2)=0.                                                       ,20
      C(3)=0.                                                       ,21
      C(4)=-XCEPT(J)                                                ,22
      CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                    ,23
      IF (NINT) 250,260,7                                           ,24
 7    XCOR(I,J)=Q(1,1)                                              ,25
      YCOR(I,J)=Q(2,1)                                              ,26
 10   ZCOR(I,J)=Q(3,1)

C     DEFINE KSTART    KSTART=1 INDICATES BODY HAS
C     NO FORE CLOSING                                               ,27   ,28   ,29
 13   KSTART=1

C     DEFINE KEND     KEND=NPLN1 INDICATES BODY HAS
```

```
        CRNRB                                                                      08/17/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C        NO AFT CLOSING                                                  ,30
         KEND=NPLN1

C        REDEFINE X-, Y- AND Z-COORDINATES OF BODY PANEL
C        CORNER POINTS FOR BODY PANELS IN BODY-WING
C        INTERSECTION REGION                                             ,31
     27  IF (KOPTB) 300,30,110                                           ,32
     30  IF (NPLNW) 300,110,35                                           ,33
     35  IF (KINT) 300,300,40                                            ,34
     40  NPLNW1=NPLNW-1

C        CHECK IF BODY-WING INTERSECTION POINT IS
C        COINCIDENT (WITHIN A TOLERANCE) WITH A
C        BODY MERIDIAN LINE                                              ,35
         DO 50 J=1,NPLNW                                                 ,36    ,37
         DO 47 I=1,NPER                                                  ,38    ,39
         K=JLEAD+J                                                       ,40
         IF (ZCOR(I,K)-(ZCEPTW(J)+EPS/2.)) 43,43,47                      ,41
   43    ISAVE(J)=I-1                                                    ,42
         GO TO 50                                                        ,43
   47    CONTINUE                                                        ,44    ,45
         GO TO 270                                                       ,46
   50    CONTINUE                                                        ,47    ,48
         DO 55 J=1,NPLNW1                                                ,49    ,50
         IF (ISAVE(J+1)-ISAVE(J)) 280,55,280                             ,51
   55    CONTINUE                                                        ,52    ,53
         IMID=ISAVE(1)

C        CHECK IF BODY-WING INTERSECTION LINE IS
C        COINCIDENT (WITHIN A TOLERANCE) WITH A
C        SINGLE BODY MERIDIAN LINE                                       ,54
   60    DO 65 J=1,NPLNW                                                 ,55    ,56
         K=JLEAD+J                                                       ,57
         IF (ABS(ZCOR(IMID+1,K)-ZCEPTW(J))-EPS) 65,65,70                 ,58
   65    CONTINUE                                                        ,59    ,60
         GO TO 110                                                       ,61
   70    NPER=NPER+1                                                     ,62
         NPER1=NPER-1                                                    ,63
         ISTART=IMID+2                                                   ,64
         DO 75 I=ISTART,NPER                                             ,65    ,66
         DO 75 J=1,NPLANE                                                ,67    ,68
         L=NPER-(I-ISTART)                                               ,69
         K=L-1                                                           ,70
         XCOR(L,J)=XCOR(K,J)                                             ,71
         YCOR(L,J)=YCOR(K,J)                                             ,72
   75    ZCOR(L,J)=ZCOR(K,J)                                             ,73    ,74    ,75
         I=IMID+1

C        DEFINE ADDITIONAL MERIDIAN LINE
C        AVERAGE ADJACENT POINTS FORE AND AFT OF
C        INTERSECTION REGION                                             ,76
         DO 80 J=1,NPLANE                                                ,77    ,78
         XCOR(I,J)=XCOR(I-1,J)+.5*(XCOR(I+1,J)-XCOR(I-1,J))              ,79
         YCOR(I,J)=YCOR(I-1,J)+.5*(YCOR(I+1,J)-YCOR(I-1,J))              ,80
```

```
 80    ZCOR(I,J)=ZCOR(I-1,J)+.5*(ZCOR(I+1,J)-ZCOR(I-1,J))

C      USE BODY-WING INTERSECTION POINTS IN BODY-
C      WING INTERSECTION REGION                                          ,81   ,82
       DO 90 J=1,NPLNW                                                   ,83   ,84
       K=JLEAD+J                                                         ,85
       XCOR(I,K)=XCEPTW(J)                                               ,86
       YCOR(I,K)=YCEPTW(J)                                               ,87
  90   ZCOR(I,K)=ZCEPTW(J)                                               ,88   ,89
 100   GO TO 110

C      CALCULATE ADDITIONAL PANEL POINTS     XINT,YINT,
C      ZINT ARE POINTS ON PANEL LEADING AND TRAILING
C      EDGES USED TO DEFINE (STREAMWISE) PANEL PARTS
C      KPANEL IS CODE DEFINING PANEL TYPE                                ,90
 110   DO 220 I=1,NPERI                                                  ,91   ,92
       DO 220 J=1,NPLN1                                                  ,93   ,94
       KPANEL(I,J)=1                                                     ,95
       P(1,1)=XCOR(I,J+1)                                                ,96
       P(2,1)=YCOR(I,J+1)                                                ,97
       P(3,1)=ZCOR(I,J+1)                                                ,98
       P(1,2)=XCOR(I+1,J+1)                                              ,99
       P(2,2)=YCOR(I+1,J+1)                                              ,100
       P(3,2)=ZCOR(I+1,J+1)                                              ,101
       R(1,1)=XCOR(I,J)                                                  ,102
       R(2,1)=YCOR(I,J)                                                  ,103
       R(3,1)=ZCOR(I,J)                                                  ,104
       R(1,2)=XCOR(I+1,J)                                                ,105
       R(2,2)=YCOR(I+1,J)                                                ,106
       R(3,2)=ZCOR(I+1,J)

C      LOCATE INBOARD POINT                                              ,107
       IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 153,153,155                 ,108
 153   C(1)=0.                                                           ,109
       C(2)=-1.                                                          ,110
       C(3)=0.                                                           ,111
       C(4)=YCOR(I,J)                                                    ,112
       GO TO 157                                                         ,113
 155   C(1)=0.                                                           ,114
       C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1))     ,115
       C(3)=-1.                                                          ,116
       C(4)=ZCOR(I,J)-C(2)*YCOR(I,J)

C      ATTEMPT TRAILING-EDGE INTERSECTION                                ,117
 157   CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                             ,118
       IF (NINT) 250,170,160                                             ,119
 160   XINT(I,J,1)=Q(1,1)                                                ,120
       YINT(I,J,1)=Q(2,1)                                                ,121
       ZINT(I,J,1)=Q(3,1)                                                ,122
       GO TO 190                                                         ,123
 170   KPANEL(I,J)=KPANEL(I,J)+1                                         ,124
       IF (ABS(ZCOR(I+1,J)-ZCOR(I,J))-F) 173,173,175                     ,125
 173   D(1)=0.                                                           ,126
       D(2)=-1.                                                          ,127
```

```
      CRNRB                                                          08/17/65
   EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      D(3)=0.                                                        ,128
      D(4)=YCOR(I,J+1)                                               ,129
      GO TO 177                                                      ,130
  175 D(1)=0.                                                        ,131
      D(2)=-(YCOR(I+1,J)-YCOR(I,J))/(ZCOR(I+1,J)-ZCOR(I,J))          ,132
      D(3)=-1.                                                       ,133
      D(4)=ZCOR(I,J+1)-D(2)*YCOR(I,J+1)

C     ATTEMPT LEADING-EDGE INTERSECTION                              ,134
  177 CALL POLXN (D,R,M,N,F,MAX,Q,LOC,NINT)                          ,135
      IF (NINT) 250,260,180                                          ,136
  180 XINT(I,J,1)=Q(1,1)                                             ,137
      YINT(I,J,1)=Q(2,1)                                             ,138
      ZINT(I,J,1)=Q(3,1)

C     LOCATE OUTBOARD POINT                                          ,139
  190 IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 193,193,195              ,140
  193 C(1)=0.                                                        ,141
      C(2)=-1.                                                       ,142
      C(3)=0.                                                        ,143
      C(4)=YCOR(I+1,J)                                               ,144
      GO TO 197                                                      ,145
  195 C(1)=0.                                                        ,146
      C(2)=-(YCOR(I+1,J+1)-YCOR(I,J+1))/(ZCOR(I+1,J+1)-ZCOR(I,J+1))  ,147
      C(3)=-1.                                                       ,148
      C(4)=ZCOR(I+1,J)-C(2)*YCOR(I+1,J)

C     ATTEMPT TRAILING-EDGE INTERSECTION                             ,149
  197 CALL POLXN (C,P,M,N,F,MAX,Q,LOC,NINT)                          ,150
      IF (NINT) 250,210,200                                          ,151
  200 XINT(I,J,2)=Q(1,1)                                             ,152
      YINT(I,J,2)=Q(2,1)                                             ,153
      ZINT(I,J,2)=Q(3,1)                                             ,154
      GO TO 220                                                      ,155
  210 KPANEL(I,J)=KPANEL(I,J)+2                                      ,156
      IF (ABS(ZCOR(I+1,J)-ZCOR(I,J))-F) 213,213,215                  ,157
  213 D(1)=0.                                                        ,158
      D(2)=-1.                                                       ,159
      D(3)=0.                                                        ,160
      D(4)=YCOR(I+1,J+1)                                             ,161
      GO TO 217                                                      ,162
  215 D(1)=0.                                                        ,163
      D(2)=-(YCOR(I+1,J)-YCOR(I,J))/(ZCOR(I+1,J)-ZCOR(I,J))          ,164
      D(3)=-1.                                                       ,165
      D(4)=ZCOR(I+1,J+1)-D(2)*YCOR(I+1,J+1)

C     ATTEMPT LEADING-EDGE INTERSECTION                              ,166
  217 CALL POLXN (D,R,M,N,F,MAX,Q,LOC,NINT)                          ,167
      IF (NINT) 250,260,218                                          ,168
  218 XINT(I,J,2)=Q(1,1)                                             ,169
      YINT(I,J,2)=Q(2,1)                                             ,170
      ZINT(I,J,2)=Q(3,1)                                             ,171
  220 CONTINUE                                                       ,172   ,173   ,174
      GO TO 300
```

```
C      ERROR MESSAGES                                                   ,175
 250   WRITE (NWRITE,900)                                               ,176   ,177
       GO TO 295                                                        ,178
 260   WRITE (NWRITE,910)                                               ,179   ,180
       GO TO 295                                                        ,181
 270   WRITE (NWRITE,920)                                               ,182   ,183
       GO TO 295                                                        ,184
 280   IMAX=ISAVE(1)                                                    ,185
       DO 285 J=2,NPLNW                                                 ,186   ,187
 285   IMAX=MAXO(IMAX,ISAVE(J))                                         ,188   ,189
       WRITE (NWRITE,930) IMAX                                          ,190   ,191   ,192
       GO TO 295                                                        ,193
 295   KODEB=1                                                          ,194
       GO TO 300                                                        ,195
 300   RETURN                                                           ,196
 900   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
      140HPANEL (PART) CORNER POINT NOT CALCULATED/10X,
      124HSUBROUTINE POLXN FAILURE/10X,
      152HINTERSECTION NOT CALCULATED DUE TO ZERO COEFFICIENTS)
 910   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
      140HPANEL (PART) CORNER POINT NOT CALCULATED/10X,
      124HSUBROUTINE POLXN FAILURE/10X,
      127HINTERSECTION DOES NOT EXIST)
 920   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
      156HBODY-(PLANAR) WING INTERSECTION NOT CORRECTLY CALCULATED)
 930   FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRB/10X,
      140HBODY-(PLANAR) WING INTERSECTION CROSSES ,I2,
      221HTH BODY MERIDIAN LINE)
        END                                                             ,197
```

EXTERNAL FORMULA NUMBER - SOURCE STATEMENT - INTERNAL FORMULA NUMBER(S)

```
      SUBROUTINE CRNRW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(4),P(3,50),Q(3,5),LOC(3)
C     SUBROUTINE TO CALCULATE WING PANEL CORNER
C     POINT COORDINATES
C     CALCULATE Y-AXIS INTERCEPT OF OUTBOARD CUTTING
C     PLANE
      IF (KOPTW) 300,10,80                                     ,1
   10 IF (IJ) 300,13,300                                       ,2
   13 IF (SLOPE) 20,14,20                                      ,3
   14 IF (KPNT) 17,17,18                                       ,4
   17 YCEPT(NPLANE)=VALUE(2)                                   ,5
      GO TO 27                                                 ,6
   18 YCEPT(NPLANE)=VALUE(4)                                   ,7
      GO TO 27                                                 ,8
   20 IF (KPNT-1) 23,23,25                                     ,9
   23 YCEPT(NPLANE)=VALUE(2)-VALUE(1)/SLOPE                    ,10
      GO TO 27                                                 ,11
   25 YCEPT(NPLANE)=VALUE(4)-VALUE(3)/SLOPE                    ,12
   27 IF (KINT) 300,40,30
C     DEFINE INBOARD EDGE PANEL CORNER POINTS FOR
C     INBOARD COLUMN OF PANELS TO BE BODY-WING
C     INTERSECTION POINTS IF BODY AND WING INTER-
C     SECT     OTHERWISE SKIP TO STATEMENT 40                  ,13
   30 DO 35 J=1,NPER                                           ,14
      XCOR(1,J)=XI(J)                                          ,15
      YCOR(1,J)=YI(J)                                          ,16
   35 ZCOR(1,J)=ZI(J)
C     CALCULATE X-, Y- AND Z-COORDINATES OF PANEL
C     CORNER POINTS     PREVIOUSLY DETERMINED
C     WING PERCENT LINES ARE INTERSECTED WITH
C     SERIES OF CUTTING PLANES                                 ,17    ,18
   40 DO 70 J=1,NPER                                           ,19
      NPOINT=NPTS(J)                                           ,20
      DO 43 I=1,NPOINT                                         ,21
      P(1,I)=X(I,J)                                            ,22
      P(2,I)=Y(I,J)                                            ,23
   43 P(3,I)=Z(I,J)                                            ,24    ,25
      DO 50 I=1,NPLN1                                          ,26
```

```
      K=I+KINT                                                  ,27
      C(1)=0.                                                   ,28
      C(2)=1.                                                   ,29
      C(3)=0.                                                   ,30
      C(4)=-YCEPT(I)                                            ,31
      E=0.001                                                   ,32
      M=3                                                       ,33
      MAX=1                                                     ,34
      CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                ,35
      IF (NINT) 230,240,47                                      ,36
 47   XCOR(K,J)=Q(1,1)                                          ,37
      YCOR(K,J)=Q(2,1)                                          ,38
 50   ZCOR(K,J)=Q(3,1)                                          ,39   ,40
      IF (SLOPE) 57,53,57                                       ,41
 53   C(4)=-YCEPT(NPLANE)                                       ,42
      GO TO 60                                                  ,43
 57   C(1)=1.                                                   ,44
      C(2)=-SLOPE                                               ,45
      C(3)=0.                                                   ,46
      C(4)=SLOPE*YCEPT(NPLANE)                                  ,47
 60   CALL POLXN (C,P,M,NPOINT,E,MAX,Q,LOC,NINT)                ,48
      IF (NINT) 230,240,65                                      ,49
 65   L=NPLANE+KINT                                             ,50
      XCOR(L,J)=Q(1,1)                                          ,51
      YCOR(L,J)=Q(2,1)                                          ,52
      ZCOR(L,J)=Q(3,1)                                          ,53
 70   CONTINUE                                                  ,54   ,55
 75   NPLANE=NPLANE+KINT                                        ,56
      NPLN1=NPLANE-1

C     DEFINE KSTART       IF KSTART = 1,WING-BODY
C     INTERSECTION IS STRAIGHT LINE IN STREAMWISE
C     DIRECTION     IF KSTART = 2,OTHERWISE                     ,57
 80   DO 85 J=1,NPER1                                           ,58
      IF (ABS(YCOR(1,J+1)-YCOR(1,J))-.001) 85,85,87             ,59
 85   CONTINUE                                                  ,60   ,61
      KSTART=1                                                  ,62
      GO TO 100                                                 ,63
 87   KSTART=2

C     DEFINE KEND       IF KEND EQUAL TO NPLN1
C     WING HAS STREAMWISE OUTBOARD EDGE       IF
C     KEND EQUAL TO NPLN1 OTHERWISE                             ,64
 100  DO 110 J=1,NPER1                                          ,65
      IF (ABS(YCOR(NPLANE,J+1)-YCOR(NPLANE,J))-.001) 110,110,120 ,66
 110  CONTINUE                                                  ,67   ,68
      KEND=NPLN1                                                ,69
      GO TO 130                                                 ,70
 120  KEND=NPLN1-1

C     CALCULATE ADDITIONAL PANEL POINTS FOR
C     PANELS IN BODY-WING INTERSECTION
C     XINT,YINT,ZINT ARE POINTS ON PANEL LEADING AND
C     TRAILING EDGES USED TO DEFINE (STREAMWISE) PANEL
```

```
C        PARTS                                                              ,71
  130    IF (KSTART-1) 300,170,140                                          ,72
  140    DO 160 J=1,NPER1                                                   ,73
         P(1,1)=XCOR(2,J+1)                                                 ,74
         P(2,1)=YCOR(2,J+1)                                                 ,75
         P(3,1)=ZCOR(2,J+1)                                                 ,76
         P(1,2)=XCOR(1,J+1)                                                 ,77
         P(2,2)=YCOR(1,J+1)                                                 ,78
         P(3,2)=ZCOR(1,J+1)
C        LOCATE POINT                                                       ,79
         C(1)=0.                                                            ,80
         C(2)=-1.                                                           ,81
         C(3)=0.                                                            ,82
         C(4)=YCOR(1,J)                                                     ,83
         E=0.001                                                            ,84
         M=3                                                                ,85
         MAX=1                                                              ,86
         N=2
C        ATTEMPT TRAILING-EDGE INTERSECTION                                 ,87
         CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                              ,88
         IF (NINT) 230,150,145                                              ,89
  145    XINT(1,J)=Q(1,1)                                                   ,90
         YINT(1,J)=Q(2,1)                                                   ,91
         ZINT(1,J)=Q(3,1)                                                   ,92
         KPANEL(1,J)=1                                                      ,93
         GO TO 160                                                          ,94
  150    P(1,1)=XCOR(2,J)                                                   ,95
         P(2,1)=YCOR(2,J)                                                   ,96
         P(3,1)=ZCOR(2,J)                                                   ,97
         P(1,2)=XCOR(1,J)                                                   ,98
         P(2,2)=YCOR(1,J)                                                   ,99
         P(3,2)=ZCOR(1,J)                                                   ,100
         C(4)=YCOR(1,J+1)                                                   ,101
         CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                              ,102
         IF (NINT) 230,240,155                                              ,103
  155    XINT(1,J)=Q(1,1)                                                   ,104
         YINT(1,J)=Q(2,1)                                                   ,105
         ZINT(1,J)=Q(3,1)                                                   ,106
         KPANEL(1,J)=2                                                      ,107
  160    CONTINUE                                                           ,108  ,109
         DO 165 J=1,NPER1                                                   ,110
         IF (YINT(1,J)-YCOR(2,J)) 165,165,250                               ,111
  165    CONTINUE                                                           ,112  ,113
  170    IF (KEND-NPLN) 180,210,300                                         ,114
  180    DO 200 J=1,NPER1                                                   ,115
         P(1,1)=XCOR(NPLN1,J)                                               ,116
         P(2,1)=YCOR(NPLN1,J)                                               ,117
         P(3,1)=ZCOR(NPLN1,J)                                               ,118
         P(1,2)=XCOR(NPLANE,J)                                              ,119
         P(2,2)=YCOR(NPLANE,J)                                              ,120
         P(3,2)=ZCOR(NPLANE,J)                                              ,121
         C(1)=0.                                                            ,122
```

```
      C(2)=-1.                                                      ,123
      C(3)=0.                                                       ,124
      C(4)=YCOR(NPLANE,J+1)                                         ,125
      E=0.001                                                       ,126
      M=3                                                           ,127
      MAX=1                                                         ,128
      N=2                                                           ,129
      CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                         ,130
      IF (NINT) 230,190,185                                         ,131
 185  XINT(2,J)=Q(1,1)                                              ,132
      YINT(2,J)=Q(2,1)                                              ,133
      ZINT(2,J)=Q(3,1)                                              ,134
      KPANEL(2,J)=1                                                 ,135
      GO TO 200                                                     ,136
 190  P(1,1)=XCOR(NPLN1,J+1)                                        ,137
      P(2,1)=YCOR(NPLN1,J+1)                                        ,138
      P(3,1)=ZCOR(NPLN1,J+1)                                        ,139
      P(1,2)=XCOR(NPLANE,J+1)                                       ,140
      P(2,2)=YCOR(NPLANE,J+1)                                       ,141
      P(3,2)=ZCOR(NPLANE,J+1)                                       ,142
      C(4)=YCOR(NPLANE,J)
C     ATTEMPT LEADING-EDGE INTERSECTION                             ,143
      CALL POLXN (C,P,M,N,E,MAX,Q,LOC,NINT)                         ,144
      IF (NINT) 230,240,195                                         ,145
 195  XINT(2,J)=Q(1,1)                                              ,146
      YINT(2,J)=Q(2,1)                                              ,147
      ZINT(2,J)=Q(3,1)                                              ,148
      KPANEL(2,J)=2                                                 ,149
 200  CONTINUE                                                      ,150   ,151
      DO 205 J=1,NPER1                                              ,152
      IF (YINT(2,J)-YCOR(NPLN1,J)) 260,205,205                      ,153
 205  CONTINUE                                                      ,154   ,155
 210  IF (KOPTW) 300,300,270
C     ERROR MESSAGES                                                ,156
 230  WRITE (NWRITE,900)                                            ,157   ,158
      GO TO 265                                                     ,159
 240  WRITE (NWRITE,910)                                            ,160   ,161
      GO TO 265                                                     ,162
 250  WRITE (NWRITE,920)                                            ,163   ,164
      GO TO 265                                                     ,165
 260  WRITE (NWRITE,930)                                            ,166   ,167
      GO TO 265                                                     ,168
 265  KODEW=1                                                       ,169
      GO TO 300                                                     ,170
 270  IF (KEND-NPLN1) 280,290,300                                   ,171
 280  J=NPLANE                                                      ,172
      SLOPE=(XCOR(NPER,J)-XCOR(1,J))/(YCOR(NPER,J)-YCOR(1,J))       ,173
      GO TO 300                                                     ,174
 290  SLOPE=0.                                                      ,175
 300  RETURN                                                        ,176
 900  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRW/10X,
     141HPANEL (PART-) CORNER POINT NOT CALCULATED/10X,
```

```
     CRNRW                                                          09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

    124HSUBROUTINE POLXN FAILURE/10X,
    152HINTERSECTION NOT CALCULATED DUE TO ZERO COEFFICIENTS)
910  FORMAT (1H1,9X,32HERROR MESSAGE - SUBROUTINE CRNRW/10X,
    141HPANEL (PART-) CORNER POINT NOT CALCULATED/10X,
    124HSUBROUTINE POLXN FAILURE/10X,
    127HINTERSECTION DOES NOT EXIST)
920  FORMAT (1H1,9X,1CHDATA ERROR/10X,
    135HINCORRECT WING PANEL (FIRST COLUMN))
930  FORMAT (1H1,9X,1GHDATA ERROR/10X,
    134HINCORRECT WING PANEL (LAST COLUMN))
     END                                                            ,177
```

```
                                                                05/22/67
     CUBIC2     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C
C
      SUBROUTINE CUBIC2(X,Y,D,C,J)
C
      DIMENSION X(1),Y(1),D(1),C(1)
C
      X2=X(2)
      B=X-X2
      IF (B)10,5,10
    5 J=3
      GC TO 30
   10 CALL OVERFL(J)                                               6
      A=(Y-Y(2))/B
      E=X+X2
      C(4)=(D+D(2)-A-A)/B/B
      C(3)=(A-D(2))/B-C(4)*(E+X2)
      C(2)=A-E*C(3)-C(4)*(E*X2+X**2)
      C=Y(2)-X2*(C(2)+X2*(C(3)+X2*C(4)))
      CALL OVERFL(J)                                               8
      J=3-J
   30 RETURN
      END
```

```
      CVEL                                                                08/16/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE CVEL(M,N,NTAPEX,A,B,C,CL,U,V,W)

C     COMPUTES VELOCITY COMPONENTS FOR A GIVEN PANEL PRESSURE DIFFERENCE

      DIMENSION A(1),B(1),C(1),U(1),V(1),W(1),CL(1)

      DO 100 J=1,M                                                        ,1
      U(J)=0.0                                                            ,2
      V(J)=0.0                                                            ,3
      W(J)=0.0                                                            ,4
  100 CONTINUE
                                                                          ,5    ,6
      DO 200 J=1,N                                                        ,7
      READ (NTAPEX) (A(I),B(I),C(I),I=1,M)                                ,8    ,9    ,10   ,11   ,12
      DO 200 I=1,M                                                        ,13
      U(I)=U(I)+A(I)*CL(J)                                                ,14
      V(I)=V(I)+B(I)*CL(J)                                                ,15
      W(I)=W(I)+C(I)*CL(J)                                                ,16
  200 CONTINUE
                                                                          ,17   ,18   ,19
      RETURN                                                              ,20
      END                                                                 ,21
```

```
      DCPD                                                          08/16/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE DCPD(NM,NTAPEX,A,ALPHAM,CLM)

C     COMPUTES WING PANEL PRESSURE DIFFERENCE

      DIMENSION A(1),ALPHAM(1),CLM(1)

      DO 100 J=1,NM                                                 ,1
      CLM(J)=0.0                                                    ,2
  100 CONTINUE
                                                                    ,3   ,4
      DO 200 J=1,NM                                                  5
      READ (NTAPEX) (A(I),I=1,NM)                                   ,6   ,7   ,8   ,9   ,10
      DO 200 I=1,NM                                                 ,11
      CLM(I)=CLM(I)+A(I)*ALPHAM(J)                                  ,12
  200 CONTINUE
                                                                    ,13  ,14  ,15
      RETURN                                                        ,16
      END                                                           ,17
```

```
      SUBROUTINE DCPI (NB,NW,NTAPEX,A,ALPHAB,CLW,CLB)
C     COMPUTES BODY PANEL PRESSURE DIFFERENCE
      DIMENSION A(1),ALPHAB(1),CLW(1),CLB(1)
      DO 100 J=1,NB                                         ,1
      CLB(J)=0.                                             ,2
  100 CONTINUE
                                                            ,3    ,4
      DO 200 J=1,NB                                         ,5
      READ (NTAPEX) (A(I),I=1,NB)                           ,6    ,7    ,8    ,9    ,10
      DO 200 I=1,NB                                         ,11
      CLB(I)=CLB(I)+A(I)*ALPHAB(J)                          ,12
  200 CONTINUE                                              ,13   ,14   ,15
      DO 300 J=1,NW                                         ,16
      READ (NTAPEX) (A(I),I=1,NB)                           ,17   ,18   ,19   ,20   ,21
      DO 300 I=1,NB                                         ,22
      CLB(I)=CLB(I)+A(I)*CLW(J)                             ,23
  300 CONTINUE
                                                            ,24   ,25   ,26
      RETURN                                                ,27
      END                                                   ,28
```

```
     DED        - EFN   SOURCE STATEMENT  -  IFN(S)  -                06/02/67

SUBROUTINE DED(NARG)                                                  DEDIS009
COMMON /JDED/I                                                        DEDIS010
I=3                                                                   DEDIS011    3
CALL DEDIS(NUM,IND,NTEST,I)                                           DEDIS012
NARG=0                                                                DEDIS013
IF (I.GT.3)NARG=1                                                     DEDIS014
RETURN                                                                DEDIS015
END
```

```
      SUBROUTINE DEDIS(NUM,IND,NTEST,IFROM)                              DEDIS016
      DOUBLE PRECISION DP                                                DEDIS017
      INTEGER ALPHA,R,TSXJC5,TOTO27                                      DEDIS018
      LOGICAL SW1,SW2,DECRH,INCRH                                        DEDIS019
      EQUIVALENCE (DP,D(1)),(TEMP,    IADD,T(1),IT(1)), (TEMP2,KTEMP)    DEDIS020
      DIMENSION T(1),D(2),RKA(7,4) ,IT(1)                                DEDIS021
      DATA ((RKA(I,J),I=1,7),J=1,4)/0.,2.,-1.,3.,-.5,1.,0.,0.,2.,-.5,-1.DEDIS022
     1,.5,-.333333333,0.,-.5,1.,0.,-1.,0.,1.,0.,2.,6.,1.,3.,-.5,-3.,0./ DEDIS023
      DATA ERRN1/3HN=0/,ERRN2/5HN FLT/,ERRHB/3HH=0/,ERRE/3HE=0/,ERRI/    DEDIS024
     13HIND/                                                             DEDIS025
      GO TO (1000,2000,3000) , IFROM                                     DEDIS026
 1000 SW1=.TRUE.                                                         DEDIS027
      SW2=.TRUE.                                                         DEDIS028
      TOTO27=2**27                                                       DEDIS029    4
      IADD =  LOC(NUM)- LOC(IADD)+65537                                  DEDIS030    6    7
      INTEST =  LOC(NTEST)- LOC(IADD)+65537                              DEDIS031    9   10
      IIND= LOC(IND)- LOC(IADD)+65537                                    DEDIS032   12   13
      JND = IT(IIND)                                                     DEDIS033
      N=NUM                                                              DEDIS034
      N1=IADD+N                                                          DEDIS035
      N2=N1+N                                                            DEDIS036
      N3=N2+N                                                            DEDIS037
      N4=N3+N                                                            DEDIS038
      N5=N4+N                                                            DEDIS039
      N6=N5+N                                                            DEDIS040
      N7=N6+N                                                            DEDIS041
      N8=N7+N                                                            DEDIS042
      N9=N8+N                                                            DEDIS043
      N10=N9+N                                                           DEDIS044
      N11=N10+N                                                          DEDIS045
      N12=N11+N                                                          DEDIS046
C     TEST IF N = 0                                                      DEDIS047
      IF (N.EQ.0) GO TO 32                                               DEDIS048
C     TEST IF N IS REAL                                                  DEDIS049
      IF (N.GT.4096) GO TO 33                                            DEDIS050
C     TEST IF H=0                                                        DEDIS051
      IF (ABS(T(N12)).LT.1.E-35) GO TO 34                                DEDIS052
C     ASSURE IND=0, 1, OR 2                                              DEDIS053
      IF (((IABS(IND).NE.0).AND.(IND.NE.1)).AND.(IND.NE.2)) GO TO 36     DEDIS054
C     IND IS 0, 1, OR 2                                                  DEDIS055
      IF(IND.NE.0)GO TO 27                                               DEDIS056
C     TEST IF E=0                                                        DEDIS057
      IF (ABS(T(N12+8)).LT.1.E-35) GO TO 35                              DEDIS058
   27 IF (IND-1) 21,20,19                                                DEDIS059
   19 SW2=.FALSE.                                                        DEDIS060
      GO TO 677                                                          DEDIS061
   20 SW1=.FALSE.                                                        DEDIS062
      GO TO 677                                                          DEDIS063
   21 IF (T(N12+3)) 9001,9000,9001                                       DEDIS064
 9000 BETA=.5                                                            DEDIS065
      GO TO 9002                                                         DEDIS066
 9001 BETA=T(N12+3)                                                      DEDIS067
 9002 EBAR=14.*T(N12+8)                                                  DEDIS068
```

```
                                                                       DEDIS123
      TSXJC5=2                                                         DEDIS124
      GO TC 999                                                        DEDIS125
12    IF (ALPHA-1) 900,901,902                                         DEDIS126
902   DC 905 I=1,N                                                     DEDIS127
      N2QI = N2-I                                                      DEDIS128
      N11QI = N11-I                                                    DEDIS129
  905 T(N2QI)=T(N11QI)                                                 DEDIS130
      GO TO 2212                                                       DEDIS131
901   DC 904 I=1,N                                                     DEDIS132
      N3QI = N3-I                                                      DEDIS133
      N11QI = N11-I                                                    DEDIS134
  904 T(N3QI)=T(N11QI)                                                 DEDIS135
      GO TO 2212                                                       DEDIS136
 900  DO 903 I=1,N                                                     DEDIS137
      N4QI = N4-I                                                      DEDIS138
      N11QI = N11-I                                                    DEDIS139
      T(N4QI)=T(N11QI)                                                 DEDIS140
      N6QI = N6-I                                                      DEDIS141
      N12QI = N12-I                                                    DEDIS142
      T(N6QI)=T(N12QI)                                                 DEDIS143
      N5QI = N5-I                                                      DEDIS144
      N10QI = N10-I                                                    DEDIS145
 903  T(N5QI)=T(N10QI)                                                 DEDIS146
2212  ALPHA=ALPHA+1                                                    DEDIS147
 23   RKA(7,1)=T(N12)/2.                                               DEDIS148
      RKA(7,3)=RKA(7,1)                                                DEDIS149
      NPASS=1                                                          DEDIS150
      GO TO 7023                                                       DEDIS151
3000  GO TO (26,3,24,4008,6) , TSXJC5                                  DEDIS152
26    DO 3026 I=1,N                                                    DEDIS153
      N10QI = N10-I                                                    DEDIS154
 3026 T(N10QI)=0.                                                      DEDIS155
      ALPHA=0                                                          DEDIS156
      R=0                                                              DEDIS157
      IT(INTEST)=1                                                     DEDIS158
      RETURN                                                           DEDIS159
3     ENP1=0.                                                          DEDIS160
      DO 3003 I=1,N                                                    DEDIS161
      N8QI = N8-I                                                      DEDIS162
      D(1)=T(N8QI)                                                     DEDIS163
      N10QI = N10-I                                                    DEDIS164
      D(2)=T(N10QI)                                                    DEDIS165
      N3QI = N3-I                                                      DEDIS166
      N2QI = N2-I                                                      DEDIS167
      N1QI = N1-I                                                      DEDIS168
      N11QI = N11-I                                                    DEDIS169
      TEMP=T(N12)*((1./24.)*T(N3QI)-(5./24.)*T(N2QI)+(19./24.)*T(N1QI)+(DEDIS170
     13./8.)*T(N11QI))                                                 DEDIS171
      DP=DP+TEMP                                                       DEDIS172
      T(N10QI)=D(2)                                                    DEDIS173
      N12QI = N12-I                                                    DEDIS174
      T(N12QI)=D(1)                                                    DEDIS175
      N9QI = N9-I                                                      DEDIS176
 3003 ENP1=AMAX1(ENP1,(ABS(T(N9QI)-T(N12QI))/AMAX1(A,ABS(T(N12QI)))))
```

```
 4      IF (.NOT.SW2)      GO TO 5                                        DEDIS177
        IF (EBAR-ENP1)     9017,9017,9013                                 DEDIS178
 9017   IF (.NOT.DECRH)        GO TO 7                                    DEDIS179
        IF (ABS(T(N12)).LT.HMINDB)      GO TO 14                          DEDIS180
 7      TEMP=T(N12)                                                       DEDIS181
        T(N12)=T(N12)*BETA                                                DEDIS182
        IF (ALPHA.EQ.3)        GO TO 8                                    DEDIS183
        T(N12+1)=T(N12+1)-TEMP                                            DEDIS184
        DC 3008 I=1,N                                                     DEDIS185
        N11QI = N11-I                                                     DEDIS186
        N1QI = N1-I                                                       DEDIS187
        T(N11QI)=T(N1QI)                                                  DEDIS188
        N12QI = N12-I                                                     DEDIS189
        N8QI = N8-I                                                       DEDIS190
        T(N12QI)=T(N8QI)                                                  DEDIS191
        N10QI = N10-I                                                     DEDIS192
        N7QI = N7-I                                                       DEDIS193
 3008   T(N10QI)=T(N7QI)                                                  DEDIS194
 4008   DO 5008 I=1,N                                                     DEDIS195
        N10QI = N10-I                                                     DEDIS196
 5008   T(N10QI)=-3.*T(N10QI)                                             DEDIS197
 6008   ALPHA=0                                                           DEDIS198
        R=0                                                               DEDIS199
C       IF P-C MODE CHECK IF INTEGRATION IS GOOD POINT                    DEDIS200
      6 IF (IABS(JND).NE.0) GO TO 31                                      DEDIS201
C       IND=0 P-C MODE                                                    DEDIS202
        IF (ALPHA.NE.0) GO TO 30                                          DEDIS203
C       AEPHA = 0                                                         DEDIS204
        IF (ABS(T(N12)).GT.HINT) GO TO 31                                 DEDIS205
C       H DECREASED                                                       DEDIS206
        GO TO 900                                                         DEDIS207
C       ALPHA NOT 0                                                       DEDIS208
                                                                          DEDIS209
                                                                          DEDIS210
                                                                          DEDIS211
                                                                          DEDIS212
                                                                          DEDIS213
                                                                          DEDIS214
     30 IF(ALPHA.LE.3) GO TO 38
C       SUCCESS FROM D.E. ALPHA GREATER THAN 3 FOR P-C MODE               DEDIS216
     31 IFROM = 4                                                         DEDIS217
        RETURN                                                            DEDIS218
     38 IFROM=5
        RETURN
 9013   IF (EBARDM-ENP1) 14,14,9018                                       DEDIS219
 9018   IF (.NOT.INCRH)      GO TO 15                                     DEDIS220
 9      IF (ABS(T(N12)).GT.HMAXTB)      GO TO 5                           DEDIS221
 15     R=R+1                                                             DEDIS222
        IF (R.LT.3)      GO TO 5                                          DEDIS223
 10     T(N12)=T(N12)/BETA                                                DEDIS224
        TSXJC5=4                                                          DEDIS225
        GO TO 999                                                         DEDIS226
 14     R=0                                                               DEDIS227
 5      DO 3005 I=1,N                                                     DEDIS228
```

```
      DEDIS      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      N4QI = N4-I                                                       DEDIS229
      N3QI = N3-I                                                       DEDIS230
      N2QI = N2-I                                                       DEDIS231
      N1QI = N1-I                                                       DEDIS232
      T(N4QI)=T(N3QI)                                                   DEDIS233
      T(N3QI)=T(N2QI)                                                   DEDIS234
3005  T(N2QI)=T(N1QI)                                                   DEDIS235
      ALPHA=ALPHA+1                                                     DEDIS236
      TSXJC5=5                                                          DEDIS237
999   IT(INTEST)=0                                                      DEDIS238
      RETURN                                                            DEDIS239
24    IF (NPASS.EQ.4)        GO TO 6                                    DEDIS240
      NPASS=NPASS+1                                                     DEDIS241
7023  DO 4023 I=1,N                                                     DEDIS242
      N9QI = N9-I                                                       DEDIS243
      N11QI = N11-I                                                     DEDIS244
      T(N9QI)=RKA(1,NPASS)*T(N9QI)+T(N12)*T(N11QI)                      DEDIS245
      N10QI = N10-I                                                     DEDIS246
      TEMP=T(N9QI)/RKA(2,NPASS)+RKA(3,NPASS)*T(N10QI)                   DEDIS247
      IF (TEMP)      9014,3023,9014                                     DEDIS248
9014  JTEMP=IADD/TOTO27                                                 DEDIS249
      N12QI = N12-I                                                     DEDIS250
      IF (T(N12QI))  9015,3023,9015                                     DEDIS251
9015  TEMP2=ABS(T(N12QI))                                               DEDIS252
      JTEMP=(KTEMP/TOTO27)-IABS(JTEMP)                                  DEDIS253
      IF (JTEMP)     3023,3023,9016                                     DEDIS254
9016  IF (JTEMP-27) 9019, 9019, 9020                                    DEDIS255
9020  TEMP = 0.                                                        DEDIS256
      GO TO 3023                                                        DEDIS257
9019  JTEMP = 2**JTEMP                                                  DEDIS258
      IADD=IADD/JTEMP                                                   DEDIS259
      IADD=JTEMP*IADD                                                   DEDIS260
3023  T(N12QI)=T(N12QI)+TEMP                                            DEDIS261
4023  T(N10QI)=T(N10QI)*RKA(6,NPASS)+TEMP*RKA(4,NPASS)+T(N9QI)*RKA(5,NPADEDIS262
     1SS)                                                               DEDIS263
      T(N12+1)=T(N12+1)+RKA(7,NPASS)                                    DEDIS264
      TSXJC5=3                                                          DEDIS265
      GO TO 999                                                         DEDIS266
8     T(N12+1)=T(N12+1)-4.*TEMP                                         DEDIS267
      DO 1008 I=1,N                                                     DEDIS268
      N11QI = N11-I                                                     DEDIS269
      N4QI = N4-I                                                       DEDIS270
      T(N11QI)=T(N4QI)                                                  DEDIS271
      N12QI = N12-I                                                     DEDIS272
      N6QI = N6-I                                                       DEDIS273
      T(N12QI)=T(N6QI)                                                  DEDIS274
      N10QI = N10-I                                                     DEDIS275
      N5QI = N5-I                                                       DEDIS276
1008  T(N10QI)=T(N5QI)                                                  DEDIS277
      GO TO 6008                                                        DEDIS278
```

381

```
   32 ERROR = ERRN1                                                          DEDIS279
      GO TO 37                                                               DEDIS280
   33 ERROR = ERRN2                                                          DEDIS281
      GO TO 37                                                               DEDIS282
   34 ERROR = ERRHB                                                          DEDIS283
      GO TO 37                                                               DEDIS284
   35 ERROR = ERRE                                                           DEDIS285
      GO TO 37                                                               DEDIS286
   36 ERROR = ERRI                                                           DEDIS287
   37 WRITE(6,8000) ERROR                                                    DEDIS288     430
      STOP                                                                   DEDIS289
 8000 FORMAT(51H0DIFFERENTIAL EQUATION SUBROUTINE DES INPUT ERROR  A6)      DEDIS290
      END                                                                    DEDIS291
```

```
         DEFEN1                                                       07/31/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT     -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE DEFEN1(B,J,K,LTAPE,IREC)
      DIMENSION  B(1),IREC(10),JREC(10),DUM(9)
      DATA  DUM/9*0./,JREC/0,9,8*0/
C
C     USED BY DEFEND TO WRITE DEFINITION TAPE
C
      IF (IREC(3))10,20,10
C                                                              ,1
 10   JREC=IREC                                                ,2
      JREC(3)=IREC(3)                                          ,3
      WRITE(LTAPE)JREC                                         ,4    ,5    ,6
      WRITE(LTAPE)DUM                                          ,7    ,8    ,9
      GO TO 100
C                                                              ,10
 20   WRITE(LTAPE)IREC                                         ,11   ,12   ,13
      WRITE(LTAPE)(B(I),I=J,K)                                 ,14   ,15   ,16   ,17   ,18
 100  RETURN                                                   ,19
      END                                                      ,20
```

```
      SUBROUTINE DEFEND(B,LK,LO,LTAPE,LGDEF,LGTAPE,INTWB)
      DIMENSION B(1),LK(3,9),LOK(3,9),IREC(10),KR(6),LGDEF(3,6)
      LOGICAL  LGDEF,LGTAPE
      EQUIVALENCE  (LOK(1,1),LUWPL),(LOK(2,1),MUWPL),(LOK(3,1),NUWPL),
     X(LOK(1,2),LLWPL),(LOK(2,2),MLWPL),(LOK(3,2),NLWPL),
     X(LOK(1,3),LUWPLX),(LOK(2,3),MUWPLX),(LOK(3,3),NUWPLX),
     X(LOK(1,4),LLWPLX),(LOK(2,4),MLWPLX),(LOK(3,4),NLWPLX),
     X(LOK(1,5),LBPL),(LOK(2,5),MBPL),(LOK(3,5),NBPL),
     X(LOK(1,6),LSTA),(LOK(3,6),NSTA),
     X(LOK(1,7),LWPF),(LOK(2,7),MWPF),(LOK(3,7),NWPF),
     X(LOK(1,8),LUWB),(LOK(3,8),NUWB),
     X(LOK(1,9),LLWB),(LOK(3,9),NLWB)
      DATA  IREC/10*0/
C
C     SPECIAL SUBROUTINE USED BY GEOMD
C     WRITES DEFINITION TAPE FOR USE BY PANEL LINK
C
 90   IF (LGTAPE) GO TC 200                                     ,1   ,2   ,3
      WRITE (LO,110)LTAPE                                       ,4   ,5   ,6
 110  FORMAT(50H1DID NOT WRITE DEFINITION TAPE.  LOGICAL TAPE NO.
     .3HWAS I10)
      GO TO 9000
C
C     WE ALWAYS WRITE 18 RECORDS ON TAPE LTAPE, BUT IT MAY TAKE 2 CALLS OF
C     DEFEND TO DO IT.  IF A WING-BODY INTERSECTION IS CALLED FOR, WE WRITE
C     THE BODY AND WING RECORDS BEFORE FINDING THE INTERSECTIONS BECAUSE THE
C     ORIGINAL WING STORAGE IS RE-USED.  THE REMAINING RECORDS ARE WRITTEN
C     LATER, ON A SECOND CALL.  INTWB IS A CODE WHICH CONTROLS THIS.

C     THE 3RD WORD OF ODD-NUMBERED RECORDS IS =0 IF OK, =1 IF NOT WANTED,
C     =-1 IF WANTED BUT ERROR OCCURRED.  WE WRITE DUMMY RECORDS IF NOT =0.
C     SET UP THIS WORD IN KR ARRAY.                             ,7
 200  DO 500 I=1,6                                              ,8
      IF (LGDEF(1,I).AND.LGDEF(2,I)) GO TO 400                  ,9   ,10  ,11
 300  IF (LGDEF(1,I)) GO TO 310                                 ,12  ,13  ,14
      KR(I)=1                                                   ,15
      GO TO 500                                                 ,16
 310  KR(I)=-1                                                  ,17
      GO TO 500
C                                                               ,18
 400  KR(I)=0                                                   ,19
      LGDEF(3,I)=.TRUE.                                         ,20
 500  CONTINUE
C                                                               ,21  ,22
      DO 600 J=1,9                                              ,23
      DO 600 I=1,3                                              ,24
 600  LOK(I,J)=LK(I,J)
C                                                               ,25  ,26  ,27
      IF (INTWB)650,650,2000                                    ,28
 650  REWIND LTAPE
C
C     BODY STATIONS                                             ,29
      IREC(1)=1                                                 ,30
```

```
      IREC(2)=NSTA                                                    ,31
      IREC(3)=KR(3)
C                                                                     ,32
      CALL DEFEN1(B,LSTA,LSTA+NSTA-1,LTAPE,IREC)
C
C     BASIC BODY PERCENT LINES                                        ,33
      IREC(1)=3                                                       ,34
      IREC(2)=NBPL                                                    ,35
      IREC(3)=KR(3)                                                   ,36
      IREC(4)=MBPL                                                    ,37
      CALL DEFEN1(B,LBPL,LBPL+NBPL-1,LTAPE,IREC)
C
C     WING PLANFORM (RECORDS 5,6)                                     ,38
      IREC(1)=5                                                       ,39
      IREC(2)=NWPF                                                    ,40
      IREC(3)=KR(4)                                                   ,41
      IREC(4)=MWPF                                                    ,42
      CALL DEFEN1(B,LWPF,LWPF+NWPF-1,LTAPE,IREC)
C
C     BASIC WING PERCENT LINES (RECORDS 7,8,9,10)                     ,43
      DO 1000 IA=1,2                                                  ,44
      IREC(1)=IA+IA+5                                                 ,45
      IREC(2)=LOK(3,IA)                                               ,46
      IREC(3)=KR(IA)                                                  ,47
      IREC(4)=LOK(2,IA)                                               ,48
 1000 CALL DEFEN1(B,LOK(1,IA),LOK(1,IA)+LOK(3,IA)-1,LTAPE,IREC)
C                                                                     ,49    ,50
 2000 IF (INTWB)9000,2100,2100
C
C     UPPER WING-BODY INTERSECTION POINTS (RECORDS 11,12)             ,51
 2100 IREC(1)=11                                                      ,52
      IREC(2)=NUWB                                                    ,53
      IREC(3)=KR(5)                                                   ,54
      IREC(4)=MUWPLX                                                  ,55
      CALL DEFEN1 (B,LUWB,LUWB+NUWB-1,LTAPE,IREC)
C
C     UPPER WING OUTSIDE BODY (RECORDS 13,14)                         ,56
      IREC(1)=13                                                      ,57
      IREC(2)=NUWPLX                                                  ,58
      CALL DEFEN1 (B,LUWPLX,LUWPLX+NUWPLX-1,LTAPE,IREC)
C
C     LOWER WING-BODY INTERSECTION POINTS (RECORDS 15,16)             ,59
      IREC(1)=15                                                      ,60
      IREC(2)=NLWB                                                    ,61
      IREC(3)=KR(6)                                                   ,62
      IREC(4)=MLWPLX                                                  ,63
      CALL DEFEN1 (B,LLWB,LLWB+NLWB-1,LTAPE,IREC)
C
C     LOWER WING OUTSICE BODY (RECORDS 17,18)                         ,64
      IREC(1)=17                                                      ,65
      IREC(2)=NLWPLX                                                  ,66
      CALL DEFEN1 (B,LLWPLX,LLWPLX+NLWPLX-1,LTAPE,IREC)
C                                                                     ,67
      END FILE LTAPE                                                  ,68
```

```
          DEFEND                                                          09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

9000 RETURN                                                                   ,69
     END                                                                      ,70
```

```
                                                                    06/02/67
     DEI          -  EFN   SOURCE STATEMENT  -  IFN(S)  -

SUBROUTINE DEI(NUM,IND,NTEST)                                      DEDIS005
CALL  DEDIS(NUM,IND,NTEST,2)                                       DEDIS006     2
RETURN                                                             DEDIS007
END                                                                DEDIS008
```

```
      FUNCTION DERIV1(X1,Y1,N)                                    DERI0030
C                                                                 DERI0020
C                                                                 DERI0040
      DIMENSION X(3),Y(3),X1(3),Y1(3)                             DERI0080
      EQUIVALENCE (S,K)                                           DERI0140
C                                                                 DERI0100
      DO 10 I=1,3                                                 DERI0110
      X(I)=X1(I)                                                  DERI0120
   10 Y(I)=Y1(I)                                                  DERI0130
      K=N                                                         DERI0150
C     FIND COEFFICIENTS OF THE X-SQUARED TERM                     DERI0160
C     AND THE X TERM.  NO NEED TO FIND CONSTANT                   DERI0170
C     TERM, AS IT DISAPPEARS UNDER DIFFERENTIATION.               DERI0180
      E=Y-Y(2)                                                    DERI0190
      H=Y-Y(3)                                                    DERI0200
      A=X-X(2)                                                    DERI0210
      B=X-X(3)                                                    DERI0220
C     C=X**2-X(2)**2                                              DERI0230
      C=A*(X+X(2))                                                DERI0240
C     DT=X**2-X(3)**2                                             DERI0250
      DT=B*(X+X(3))                                               DERI0260
      C3=(B*E-A*H)/(B*C-A*DT)                                     DERI0270
      C2=(E-C3*C)/A                                               DERI0280
C     TEST TO SEE WHETHER DERIVATIVE IS WANTED AT ONE             DERI0290
C     OF THE INPUT POINTS OR ELSEWHERE.                           DERI0300
      K1=IABS(K)                                                  DERI0310
      DO  20  I=1,3                                               DERI0320
      IF(K1-I)  20 ,50,20                                         DERI0330
   20 CONTINUE                                                    DERI0340
      GO TO 60                                                    DERI0350
   50 S =X(K1)                                                    DERI0360
   60 DERIV1=C2+2.0*C3*S                                          DERI0370
      RETURN                                                      DERI0380
      END                                                         DERI0390
```

```
      DERIV2      -  EFN    SOURCE STATEMENT  -  IFN(S)  -          05/22/67

      FUNCTION DERIV2(X,Y,XX)                                     DERI0030
C                                                                 DERI0020
C                                                                 DERI0040
      DIMENSION X(4),Y(4)                                         DERI0050
C                                                                 DERI0070
      DERIV2=0.0                                                  DERI0080
      IF (X(4)-X(3)) 1,7,1                                        DERI0090
    1 IF (X(4)-X(2)) 2,7,2                                        DERI0100
    2 IF (X(4)-X(1)) 3,7,3                                        DERI0110
    3 IF (X(3)-X(2)) 4,7,4                                        DERI0120
    4 IF (X(3)-X(1)) 5,7,5                                        DERI0130
    5 IF (X(2)-X(1)) 6,7,6                                        DERI0140
    6 Q41=(Y(4)-Y(1))/(X(4)-X(1))                                 DERI0150
      Q31=(Y(3)-Y(1))/(X(3)-X(1))                                 DERI0160
      Q21=(Y(2)-Y(1))/(X(2)-X(1))                                 DERI0170
      E=(Q31-Q21)/(X(3)-X(2))                                     DERI0180
      D=((Q41-Q21)/(X(4)-X(2))-E)/(X(4)-X(3))                     DERI0190
      C=E-D*(X(3)+X(2)+X(1))                                      DERI0200
      DERIV2=2.0*(C+3.0*D*XX)                                     DERI0210
    7 RETURN                                                      DERI0220
      END                                                         DERI0230


      SUBROUTINE DERIVS(XB,R,NBODYS,FD,SD)

      DIMENSION XB(1),R(1),FD(1),SD(1),C(4,50)

C     SUBROUTINE TO OBTAIN FIRST AND SECOND
C     DERIVATIVES      SUBROUTINE USES SCAMP4 TO FIT
C     A CHAIN OF CUBICS THROUGH AN ARRAY OF POINTS
C     AND TO CALCULATE FIRST DERIVATIVES AT EACH OF
C     THE POINTS      SUBROUTINE THEN CALCULATES
C     SECOND DERIVATIVES

C     FIT CUBIC CHAIN AND GET FIRST DERIVATIVES
      CALL SCAMP4(XB,R,NBODYS,-1,-1,0.,0.,C,FD,0)

C     CALCULATE SECOND DERIVATIVES                                       2
      NB1=NBODYS-1
      DO 100 I=1,NB1
  100 SD(I)=2.*C(3,I)  +6.*C(4,I)  *XB(I)
      SD(NBODYS)=2.*C(3,NB1)+6.*C(4,NB1)*XB(NBODYS)
      RETURN
      END
```

```
     DES        - EFN   SOURCE STATEMENT  - IFN(S) -                 06/02/67

C      7094(FORTRAN IV) SUBPROGRAM  IBSYS VERSION 9                 DEDIS000
       SUBROUTINE DES (NUM,IND,NTEST)                               DEDIS001
       CALL DEDIS(NUM,IND,NTEST,1)                                  DEDIS002    2
       RETURN                                                       DEDIS003
       END                                                          DEDIS004
```

```
          DISPTA                                                           07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT     -    INTERNAL FORMULA NUMBER(S)

      FUNCTION DISPTA(P,Q,J)
C
C     FIND DISTANCE FROM POINT P TO Q
C     J = DIMENSION OF P AND Q
      DIMENSION P(1),Q(1)
C
      D=0.                                                                  ,1
      DO 10 I=1,J                                                           ,2
      U=P(I)-Q(I)                                                           ,3
   10 D=D+U*U                                                               ,4      ,5
      DISPTA=SQRT (D)                                                       ,6
      RETURN                                                                ,7
      END                                                                   ,8
```

```
      SUBROUTINE DMAXL(P,N,DIST,J,U,NU)
      DIMENSION P(3,1),U(3),V(3),UC(3)
C
C     GIVEN AN ARRAY P OF N POINTS IN 3-SPACE.
C     FIND WHICH POINT IN P IS FARTHEST FROM THE LINE THRU END POINTS,
C     AND ITS DISTANCE FROM THE LINE.  IF MORE THAN ONE POINTS ARE
C     EQUALLY FAR, CHOOSE 1ST SUCH POINT IN P.
C     THE DESIRED POINT IS P(1,J)
C     NU=0 IF SUCCESS, = -1 IF N .LT. 2, =-2 IF END POINTS OF P
C       ARE COINCIDENT, =1 IF N=2.
C     INPUTS ARE P,N.  OUTPUTS ARE DIST,J,U,NU
C                                                                  ,1
      NU=0                                                         ,2
      DP=0.                                                        ,3
      J=0                                                          ,4
      IF(N-2)100,200,300                                           ,5
  100 NU=-1
      GO TO 2000                                                   ,6
C                                                                  ,7
  200 NU=1
      GO TO 2000                                                   ,8
C                                                                  ,9
  300 NM=N-1
      IF(UVECN(P,P(1,N),U,0,3))400,400,500                         ,10
C     U IS A UNIT VECTOR FROM P(1,1) TO P(1,N)                     ,11
  400 NU=-2
      GO TO 2000                                                   ,12
C
  500 DO 1000 K=2,NM                                               ,13
C                                                                  ,14
      DO 600 I=1,3
  600 V(I)=P(I,K)-P(I,1)
C     V IS A VECTOR FROM P(1,1) TO THE K-TH POINT                  ,15    ,16
C                                                                  ,17
      KE=0
      CALL VCROS(U,V,UC,D,KE)
C     UC IS A VECTOR, THE CROSS-PRODUCT U X V.                     ,18
C     D IS THE MAGNITUDE OF UC AND IS DISTANCE FROM LINE TO POINT  ,19
      IF(D-DP)1000,1000,700                                        ,20
  700 J=K
      DP=D                                                         ,21
C                                                                  ,22    ,23
 1000 CONTINUE                                                     ,24
 2000 DIST=DP                                                      ,25
 9000 RETURN                                                       ,26
      END
```

```
       ELLIPR                                                         09/14/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE ELLIPR(A,B,SIN,COS,RHO,X,Y,NU)
C
C     GIVEN A RECTANGULAR ELLIPSE WITH MAJOR, MINOR SEMI-AXES A,B
C     AND CENTER AT THE ORIGIN.
C     GIVEN THE SIN AND COS OF AN ANGLE THETA MEASURED
C     CLOCKWISE FROM THE +Y AXIS.
C
C     FIND RHO, THE LENGTH OF A RADIUS VECTOR FROM ORIGIN ALONG THETA
C     TO THE ELLIPSE.
C     FIND (X,Y), THE CORRESPONDING POINT ON THE ELLIPSE.
C
      IF(A*B)200,100,2C0                                               ,1
 100  NU=1                                                             ,2
      GO TO 9000                                                       ,3
 200  U=(SIN/A)**2+(COS/B)**2                                          ,4
      IF(U)300,300,400                                                 ,5
 300  NU=2                                                             ,6
      GO TO 9000                                                       ,7
 400  RHO=1./SQRT(U)                                                   ,8
      X=RHO*SIN                                                        ,9
      Y=RHO*COS                                                        ,10
      NU=0                                                             ,11
9000  RETURN                                                           ,12
      END                                                              ,13
```

```
      SUBROUTINE ENRYCH(A,B,NI,EPS,C,NO,NU)
      DIMENSION A(500),B(500), C(10000),D(4),E(4),F(4),G(4),W(200),Z(200ENRI0150
     1),NU(3)
C                                                                       ENRI0110
C     J. E. BROWN                      ASSEMBLED 12-15-64

C     USES OPTIM3 TO CONVERT SPARSE ARRAY A, B                          ENRI0120
C     TO DENSE ARRAY C.                                                 ENRI0130
C                                                                       ENRI0140
C     A = ARRAY OF X VALUES
C     B = ARRAY OF Y VALUES
C     NI = NUMBER OF INPUT POINTS
C     EPS = CHORD HEIGHT TOLERANCE
C     C = (X,Y,X,Y,....)
C     NO (INPUT) = LENGTH OF C
C     NO (OUTPUT) =NUMBER OF ELEMENTS PLACED IN C
C     NU = 0 IF SUCCESS
C     NU = 1 IF OPTIM3 ERROR WAS =0 (NOT POSSIBLE)
C     NU = 2 IF 4 CONSEC. POINTS NOT MONOTONIC IN EITHER X OR Y
C     NU = 3 IF OPTIM3 GENERATED 199 POINTS BETWEEN 2 GIVEN PTS AND NOT DONE.
C     NU = 4 IF LENGTH OF C (=NO) WAS TOO SMALL
C     NU = 5 IF NO LESS THAN 2*NI

C     NU(2), AN INPUT, IS THE LOGICAL NUMBER OF AN OUTPUT TAPE ON WHICH TO WRITE
C     ERROR COMMENTS, IF ANY.  NU(3), BOTH AN INPUT AND OUTPUT, IS AN ERROR
C     MESSAGE LIMITER AND COUNTER.  IF AN ERROR OCCURS FOR WHICH A MESSAGE IS
C     NORMALLY WRITTEN, NU(3) IS FIRST REDUCED BY ONE.  THE MESSAGE IS THEN
C     WRITTEN ONLY IF BOTH NU(2) AND NU(3) ARE GREATER THAN ZERO.
C                                                                       ENRI0190
      NU=0                                                                        ,1
      NOM=NO                                                                      ,2
      NO=0                                                                        ,3
      NII=NI+NI
                                                                                  ,4
      IF (NII-NOM)1,1,3                                                           ,5
    1 IF (NI-4)500,2,2                                                            ,6
    2 IF (EPS)500,500,10                                                          ,7
    3 NU=5                                                                        ,8
      GO TO 1000
                                                                                  ,9
   10 I1=1                                                              ENRI0260  ,10
      I2=1                                                              ENRI0270  ,11
      I3=1                                                              ENRI0280  ,12
      KK=1                                                              ENRI0290

                                                                                  ,13
   20 DO 320 I=I1,I2                                                    ENRI0300
                                                                                  ,14
   30 DO 35 J=1,4                                                       ENRI0310  ,15
      I4=I+J-1                                                          ENRI0320  ,16
      D(J)=A(I4)                                                        ENRI0330  ,17
   35 E(J)=B(I4)                                                        ENRI0340
                                                                                  ,18   ,19
```

```
          ENRYCH                                                              07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT      -       INTERNAL FORMULA NUMBER(S)

      L1=0                                                       ENRI0350    ,20
   40 IF (ABS(D(4)-D(1))-ABS(E(4)-E(1)))100,50,50                ENRI0360    ,21
   50 IF (D(2)-D(1))80,95,60                                     ENRI0370    ,22
   60 IF (D(3)-D(2))95,95,70                                     ENRI0380    ,23
   70 IF (D(4)-D(3))95,95,200                                    ENRI0390    ,24
   80 IF (D(3)-D(2))90,95,95                                     ENRI0400    ,25
   90 IF (D(4)-D(3))210,95,95                                    ENRI0410    ,26
   95 IF (L1)160,97,160                                          ENRI0420    ,27
   97 L1=1                                                       ENRI0430    ,28
  100 IF (E(2)-E(1))130,150,110                                  ENRI0440    ,29
  110 IF (E(3)-E(2))150,150,120                                  ENRI0450    ,30
  120 IF (E(4)-E(3))150,150,220                                  ENRI0460    ,31
  130 IF (E(3)-E(2))140,150,150                                  ENRI0470    ,32
  140 IF (E(4)-E(3))230,150,150                                  ENRI0480    ,33
  150 IF (L1)160,152,160                                         ENRI0490    ,34
  152 L1=1                                                       ENRI0500    ,35
      GO TO 50                                                   ENRI0510
C     NOT MONOTONIC IN EITHER X OR Y                             ENRI0520    ,36
  160 NU=2                                                                   ,37
      L=1                                                        ENRI0540    ,38
      GO TO 316                                                  ENRI0550
C                                                                ENRI0560    ,39
  200 DO 205 J=1,4                                               ENRI0570    ,40
      F(J)=D(J)                                                  ENRI0580    ,41
  205 G(J)=E(J)                                                  ENRI0590    ,42    ,43
      MX=1                                                       ENRI0600    ,44
      GO TO 240                                                  ENRI0610    ,45
  210 DO 215 J=1,4                                               ENRI0620    ,46
      F(J)=-D(J)                                                 ENRI0630    ,47
  215 G(J)=E(J)                                                  ENRI0640    ,48    ,49
      MX=2                                                       ENRI0650    ,50
      GO TO 240                                                  ENRI0660    ,51
  220 DO 225 J=1,4                                               ENRI0670    ,52
      F(J)=E(J)                                                  ENRI0680    ,53
  225 G(J)=D(J)                                                  ENRI0690    ,54    ,55
      MX=3                                                       ENRI0700    ,56
      GO TO 240                                                  ENRI0710    ,57
  230 DO 235 J=1,4                                               ENRI0720    ,58
      F(J)=-E(J)                                                 ENRI0730    ,59
  235 G(J)=D(J)                                                  ENRI0740    ,60    ,61
      MX=4                                                       ENRI0750    ,62
  240 CONTINUE                                                   ENRI0760
                                                                             ,63
      CALL OPTIM3 (F,G,EPS,W,Z,KK,L,MU)                          ENRI0770
C     MU = 0 IF F(1)=F(4)   (NOT POSSIBLE)
C     MU = 1 IF SUCCESS
C     MU = 2 IF 199 PTS. GENERATED AND NOT DONE                              ,64
  250 IF(MU-1)251,260,251                                        ENRI0780    ,65
  251 NU=MU+1                                                                ,66
      GO TO 316                                                  ENRI0800
                                                                             ,67
  260 IF(L-1)275,275,265                                         ENRI0810    ,68
  265 IF ((F(I3+1)-W(L))/(W(L)-W(L-1))-.2) 270,270,275                       ,69
  270 L=L-1                                                      ENRI0830    ,70
```

```
275 TA=1.0                                                        ENRI0840   ,71
    IF(MX-3)280,285,285                                           ENRI0850   ,72
280 NA=1                                                          ENRI0860   ,73
    NB=0                                                          ENRI0870   ,74
    IF(MX-2)300,290,300                                           ENRI0880   ,75
285 NA=0                                                          ENRI0890   ,76
    NB=1                                                          ENRI0900   ,77
    IF(MX-4)300,290,300                                           ENRI0910   ,78
290 TA=-1                                                         ENRI0920
                                                                             ,79
300 NOM=NOM-L-L                                                              ,80
    IF (NOM)365,305,305
                                                                             ,81
305 DO 315 J=1,L                                                             ,82
    NO=NO+2                                                       ENRI0940   ,83
    JA=NO-NA                                                      ENRI0950   ,84
    JB=NO-NB                                                      ENRI0960   ,85
    C(JA)=W(J)*TA                                                 ENRI0970   ,86
315 C(JB)=Z(J)                                                    ENRI0980
                                                                             ,87   ,88
    GO TO 320                                                     ENRI0990

C   ERROR AT 160 OR 250.  WRITE AND GO ON, WITHOUT ADDING NEW POINTS. ENRI1000 ,89
316 NU(3)=NU(3)-1                                                            ,90
    IF (NU(3))321,321,317                                                    ,91
317 LTAPE=NU(2)                                                              ,92
    IF (LTAPE)321,321,318
                                                                             ,93
318 WRITE (LTAPE,319)NU(1),EPS,NI,D,E                                        ,94   ,95   ,96
319 FORMAT(18HOENRYCH ERROR CODE I2,34H FOR THE FOUR POINTS BELOW.  EP
   1S = E12.4, 6H, NI = I4/4H X =4F20.9/4H Y =4F20.9)             ENRI1030
                                                                             ,97
321 NOM=NOM-2                                                                ,98
    IF (NOM)365,323,323                                                      ,99
323 NO=NO+1
    C(NO)=D(I3)                                                   ENRI1050   ,100
    NO=NO+1                                                       ENRI1060   ,101
    C(NO)=E(I3)                                                   ENRI1070
                                                                             ,102
320 CONTINUE                                                      ENRI1080
                                                                             ,103  ,104
330 IF (I3-2)340,350,360                                          ENRI1090   ,105
340 I2=NI-3                                                       ENRI1100   ,106
    KK=2                                                          ENRI1110   ,107
    I3=2                                                          ENRI1120   ,108
    GO TO 20                                                      ENRI1130   ,109
350 I1=I2                                                         ENRI1140   ,110
    KK=3                                                          ENRI1150   ,111
    I3=3                                                          ENRI1160   ,112
    GO TO 20                                                      ENRI1170   ,113
360 IF (NOM-2)365,37C,370                                                    ,114
365 NU=4                                                                     ,115
    GO TO 1000
                                                                             ,116
370 C(NO+1)=A(NI)                                                            ,117
```

```
          ENRYCH                                                      07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      C(NO+2)=B(NI)                                       ENRI1190   ,118
      NO=NO+2                                             ENRI1200   ,119
  420 GO TO 1000
                                                                     ,120
  500 DO 510 I=1,NI                                                  ,121
      J=I+I                                                          ,122
      C(J-1)=A(I)                                                    ,123
  510 C(J)=B(I)                                                      ,124  ,125
      NO=NII
                                                                     ,126
 1000 RETURN                                                         ,127
      END                                                 ENRI1220   ,128
```

```
      SUBROUTINE EVAL (NI,XCPT,THKW)

C     COMPUTES AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND VELOCITY
C     COMPONENTS MATRIX

      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))
                                                                   2     3     4
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      COMMON /BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     C,CHORD(210)
     I,ISYM
     N,NPART(210),NPANEL,NROW(2)
     T,THETA(210),TAIL
     U,U(210)
     V,V(210),VPM(210),VV(210),VPMM(210)
     W,W(210),WPM(210),WW(210),WPMM(210)
     X,X(210,3,4),XBAR(210),XC(210)
     Y,Y(210,3,4),YBAR(210),YC(210)
     Z,Z(210,3,4),ZBAR(210),ZC(210)

      DIMENSION AS(210),BBETAM(4),C(4),CPM(4),CPMM(4),Q(4)
     1,P(4),RPM(4),RPMM(4),UU(210),US(210),VS(210),WS(210)
     2,XCPT(210)

      EPS=0.0001
      BETA=SQRT(ABS(XMACH**2-1.))                                        7
      PI=3.1415926
      CONSTA=BETA/4.0
      CONSTB=CONSTA/PI
      CCNSTD=BETA/PI
      IT=1

C     INFLUENCING PANEL I

      GO TO (1,980,3),KACE
    1 NC=NPANEL/NROW(1)
      NR=NROW(1)
      GC TO 4
    3 NC=NWING/NROW(2)
      NR=NROW(2)
      IF (THKW.NE.1.) GO TO 4
      DC 34 I=1,NBODY
   34 XCPT(I)=XC(I)
    4 N=1
      NCT=NI-1
      DO 900 I=NI,NPANEL
    5 IF(THKW)10,40,10
   10 SL=1.0
```

```
      IT=1+I-NI-NR*(N-1)
      IF(IT)30,30,25
   25 N=N+1
      THKW=1.0
      GO TO 40
   30 THKW=-1.0
      NCT=NCT+1
      GO TO 40
   35 SL=-1.0
      THKW=1.0
   40 COST=COS(THETA(I))                                            45
      SINT=SIN(THETA(I))                                            47
      TANA= TAN(ALPHAS(I))                                          49
      ABR=BETA*TANA
      IF(ABR-1.0)50,950,950
   50 IF(ABS(ABR)-EPS)80,80,85
80    ABR=0.0
      SA=1.0
      GO TO 90
85    SA=SIGN(1.,ABR)
90    APM2=ABR*ABR
      NPT=NPART(I)
C     INFLUENCED PANEL J

      DO 850 J=1,NPANEL
      U(J)=0.0
      V(J)=0.0
      W(J)=0.0
      VPM(J)=0.
      WPM(J)=0.
      VPMM(J)=0.
      WPMM(J)=0.
      SINA=SIN(ALPHAS(J))                                           69
      COSA=COS(ALPHAS(J))                                           71
      DO 850M=1,NPT
      BBETAM(1)=(X(I,M,2)-X(I,M,1))/(BETA*((Y(I,M,2)-Y(I,M,1))*COST
     1+(Z(I,M,2)-Z(I,M,1))*SINT))
      BBETAM(3)=(X(I,M,4)-X(I,M,3))/(BETA*((Y(I,M,4)-Y(I,M,3))*COST
     1+(Z(I,M,4)-Z(I,M,3))*SINT))
      BBETAM(2)=BBETAM(1)
      BBETAM(4)=BBETAM(3)

C     CORNER POINTS K OF INFLUENCING PANEL I

      DO 810K=1,4
      B(K)=0.0
      C(K)=0.0
      Q(K)=0.0
      BPM(K)=0.
      CPM(K)=0.
      BPMM(K)=0.
      CPMM(K)=0.
      BPM=BBETAM(K)
      IF(THKW)99,99,91
```

```
   91 IF(SL)92,99,93
   92 IF(K-2)810,810,99
   93 IF(K-2)99,99,810
   99 IF(ABS(BPM)-EPS)105,105,100
  100 SM=SIGN(1.0,BPM)
      GO TC 110
  105 SM=1.0
      BPM=0.0
  110 IF (ABS(ABS(BPM)-1.0)-EPS) 115,115,120
  115 BPM=SM
  120 BPM=SM*BPM
      BPM2=BPM*BPM
      CONSTC=1.+APM2*BPM2

C     CONFIGURATION SYMMETRY CONDITION

      DC 800ISIDE=1,ISYM
      SN=(-1.0)**ISIDE                                                114
      DELTAT=THETA(I)+THETA(J)*SN
      SINDT=SIN(DELTAT)                                               117
      COSDT=COS(DELTAT)                                               118
      DELTAY=SN*(-YC(J))-Y(I,M,K)
      DELTAZ=ZC(J)-Z(I,M,K)
      XPM=XCPT(J)-X(I,M,K)
      YPM=(DELTAY*COST+DELTAZ*SINT)*SM
      ZBR=DELTAZ*COST-DELTAY*SINT
      IF (ABS(ZBR)-EPS) 140,140,145
  140 ZBR=0.0
  145 SZ=SIGN(1.,ZBR)
      ZPM2=ZBR*ZBR
      IF (ABS(YPM)-EPS) 150,150,155
  150 YPM=0.0
  155 SY=SIGN(1.,YPM)
      XIPM=XPM/BETA
      XIPM2=XIPM*XIPM
      YPM2=YPM*YPM
      IF (THKW) 700,190,700
  190 CONTINUE
      IF (ABR) 195,499,198
C     VELOCITY COMPONENTS INDUCED BY PANELS NON-PARALLEL TO X- AXIS

  195 APM=-ABR
      ZPM=-ZBR
      GC TO 199
  198 APM=ABR
      ZPM=ZBR
  199 CONTINUE
      IF (XIPM-SQRT(YPM2+ZPM2)-EPS) 200,200,250                       138
  200 IF (BPM-1./SQRT(1.-APM2)) 205,550,550                           141
  205 TERMA=SQRT(1.-BPM2*(1.-APM2))                                   144
      TERMB=ZPM-APM*BPM*YPM
      TERMC=YPM+APM*BPM*ZPM
```

```
 324 FTR1=ACOS(FTR1)*STERMC                                                         198
 325  IF (BPM-1./SQRT(TERMD))335,330,330                                             200
 330  FTR2=ACOSH((BPM*XIPM-TERMF)/SQRT(TERME))                                       203
      GO TO 340
 335 FTR2=(BPM*XIPM-TERMF)/SQRT(TERME)                                              206
      IF (FTR2+1.0) 339,339,336
 336 IF (FTR2-1.0) 337,338,338
 337 FTR2=-ACOS(FTR2)                                                               211
      GO TO 340
 338 FTR2=0.0
      GO TO 340
 339 FTR2=-PI
 340  IF (YPM) 370,345,370
 345  IF (TERMC) 355,350,350
 350  FTR3=FTR1
      GO TO 365
 355  IF (ZPM) 350,350,360
 360  FTR3=PI+FTR1
 365  FTR5=-APM2*BPM2*FTR3
      GO TO 415
 370  IF (TERMB) 400,375,400
 375  IF (TERMA) 380,390,385
 380  IF (YPM) 385,390,395
 385  FTR3=0.0
      GO TO 415
 390  FTR3=PI/2.0
      GO TO 415
 395  FTR3=PI*STERMB
      GO TO 415
 400  FTR3=((-XIPM*TERMF+BPM*(YPM2+ZPM2))/SQRT((YPM2+ZPM2)*TERME))                  240
      IF (FTR3+1.0) 395,395,405
 405  IF (FTR3-1.0) 410,385,385
 410  FTR3=ACOS (FTR3)*STERMB                                                        245
 415  FTR3=FTR3*SIGN(1.,APM)
      FTR4=SQRT(ABS(BPM2*TERMD-1.))                                                  247
      IF (YPM) 420,430,420
 420  FTR5=FTR3-FTR1*CONSTC
      IF (YPM) 430,430,422
 422  IF (TERMC) 424,430,430
 424  IF (TERMB) 426,428,430
 426  FTR5=FTR5-2.*PI*CONSTC
      GO TO 430
 428  FTR5=FTR5-PI*(2.+ABR2*BPM2)
 430  IF (YPM2+ZPM2) 435,440,435
 435  FTR6=ACOSH(XIPM/SQRT(YPM2+ZPM2))                                              261
      GO TO 445
 440  FTR6=1000.
 445  IF (TERMC) 455,450,455
 450  IF (YPM) 455,460,455
 455  FTR7=ACOSH((XIPM-APM*ZPM)/SQRT(TERMC2+TERMD*YPM2))*SQRT(TERMD)                269   270
      GO TO 465
 460  FTR7=1000.
 465  FTR8=FTR6-CONSTC*FTR7
      S=CONSTB/CONSTC*(APM*BPM*FTR4*FTR2+BPM*FTR3+FTR8/APM)
```

```
      D=CONSTB/CONSTC*(FTR4*FTR2-BPM*FTR6+FTR5/APM)*(-SA)
      P=CONSTB*FTR1
      IF (TERMC) 795,470,795
  470 P=P*SA
      GO TO 795

C     VELOCITY COMPONENTS INDUCED BY PANELS PARALLEL TO X- AXIS

  499 CONTINUE
      ZPM=ZBR
      TERMA=(XIPM-BPM*YPM)**2
      TERMB=YPM2+ZPM2
      TERMC=SQRT(TERMB)                                                     278
      TERMD=BPM2-1.0
      TERME=-TERMD
      IF (BPM2-1.0) 500,560,560
  500 IF (XIPM-TERMC)520,535,510
  510 D=CONSTB*(SQRT(TERME)*ACOS((BPM*XIPM-YPM)/SQRT
     1(TERMA+TERMD*ZPM2))+BPM*ACOSH(XIPM/TERMC)+
     2YPM/TERMB*SQRT(XIPM2-TERMB))                          283   284   285   286
      GO TO 575
  520 IF (YPM) 550,550,530
  530 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,537,535                  291
  535 IF (YPM-BPM*XIPM)550,537,540
  537 CT=.5
      GO TO 545
  540 CT=1.0
  545 D=CONSTA*SQRT(TERME)*CT                                               299
      S=CONSTA*BPM*SZ*CT
      P=CONSTA*SZ*CT
      GO TO 795
  550 D=0.0
      S=0.0
      P=0.0
      GO TO 795
  560 IF (XIPM-TERMC) 550,550,570
  570 D=CONSTB*(-SQRT(TERMD)*ACOSH((BPM*XIPM-YPM)
     1/SQRT(TERMA+TERMD*ZPM2))+BPM*ACOSH(XIPM/TERMC)
     2+YPM/TERMB*SQRT(XIPM2-TERMB))                               306   307   308
  575 IF (ZPM) 580,585,580
  580 FACTOR=     (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*
     1(TERMA+TERMD*ZPM2))                                                   312
      IF (FACTOR+1.) 600,600,582
  582 IF (FACTOR-1.) 584,590,590
  584 FACTOR=ACOS(FACTOR)*SZ                                                317
      GO TO 625
  585 IF (YPM) 590,597,595
  590 FACTOR=0.0
      GO TO 625
  595 IF (XIPM-BPM*YPM) 590,597,600
  597 FACTOR=PI/2.
      GO TO 625
  600 FACTOR=PI*SZ
  625 S=CONSTB*(BPM*FACTOR-ZPM/TERMB*SQRT(XIPM2-TERMB))                     329
      P=CONSTB*FACTOR
      GO TO 795
  700 CONTINUE
```

```
C     VELOCITY COMPONENTS INDUCED BY WING SOURCES
      ZPM=ZBR
      TERMAA=XIPM-BPM*YPM
      TERMA=(XIPM-BPM*YPM)**2
      TERMB=YPM2+ZPM2
      TERMC=SQRT(TERMB)                                                332
      TERMD=BPM2-1.0
      TERME=-TERMD
      IF (XIPM-TERMC) 704,704,708
  704 IF(TERMD)753,550,550
 708  FTR1=ACOSH(XIPM/TERMC)
      IF (TERMD) 712,716,720
  712 FTR2=ACOS((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERME)   339  340  341
      GO TO 724
  716 FTR2=SQRT(XIPM2-TERMB)/(XIPM-YPM)                                344
      GO TO 724
  720 FTR2=ACOSH((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERMD)    347  348
  724 IF (ZPM) 728,732,728
 728  FTR3=     (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*(TERMA+TERMD*ZPM2))    352
      IF (FTR3+1.0) 748,748,730
 730  IF (FTR3-1.0) 731,736,736
 731  FTR3=ACOS(FTR3)*SZ                                               357
      GO TO 750
  732 IF (YPM) 736,736,740
  736 FTR3=0.
      GO TO 750
  740 IF (XIPM-BPM*YPM) 736,744,748
  744 FTR3=PI/2.
      GO TO 750
  748 FTR3=PI*SZ
  750 IF(THKW)751,190,752
  751 P= CONSTD*(TERMAA*FTR2+YPM*FTR1-ZPM*FTR3)/CHORD(NCT)
      S=-CONSTD*(TERMAA*(FTR1-BPM*FTR2)+BPM*ZPM*FTR3
     1-SQRT(XIPM2-TERMB))/CHORD(NCT)                                   372
      D=-CONSTD*(ZPM*(BPM*FTR1-TERMD*FTR2)-TERMAA*FTR3)
     1/CHORD(NCT)
      GO TO 795
  752 P=FTR2/PI
      S=-(FTR1-BPM*FTR2)/PI
      D=FTR3/PI
      GO TO 795
  753 IF(YPM)550,550,756
  756 IF (XIPM-TERMC) 760,764,708
  760 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,768,764             383
  764 IF (YPM-BPM*XIPM) 550,768,772
  768 CT=.5
      GO TO 776
  772 CT=1.
```

```
  776 IF(THKW)777,190,778
  777 P=(TERMAA/SQRT(TERME)-ZPM*CT*SZ)*BETA/CHORD(NCT)                393
      S=BPM*P
      D=BETA*(-ZPM*SQRT(TERME)+TERMAA*CT*SZ)/CHORD(NCT)               395
      GO TO 795
  778 P=CT/SQRT(TERME)                                                399
      S=BPM*P
      D=CT*SZ


  795 CONTINUE

      SIM=1.

      R(K)=R(K)+S*SINDT*SIM
      C(K)=C(K)+D*COSDT*SM*SIM
      Q(K)=Q(K)-P*SN*SIM
      RPM(K)=RPM(K)+S*SIM
      CPM(K)=CPM(K)+D*SM*SIM
      RPMM(K)=RPMM(K)-S*SN*SIM
      CPMM(K)=CPMM(K)-D*SM*SN*SIM
 800  CONTINUE
 810  CONTINUE

      U(J)=U(J)+(Q(1)-Q(2)-Q(3)+Q(4))/BETA
      V(J)=V(J)+(R(1)-R(2)-R(3)+R(4))
      W(J)=W(J)+(C(1)-C(2)-C(3)+C(4))
      VPM(J)=VPM(J)+(RPM(1)-RPM(2)-RPM(3)+RPM(4))
      WPM(J)=WPM(J)+(CPM(1)-CPM(2)-CPM(3)+CPM(4))
      VPMM(J)=VPMM(J)+(RPMM(1)-RPMM(2)-RPMM(3)+RPMM(4))
      WPMM(J)=WPMM(J)+(CPMM(1)-CPMM(2)-CPMM(3)+CPMM(4))
 850  CONTINUE

      IF(THKW)865,855,865
  855 DO 860 J=1,NPANEL
      UU(J)=U(J)
      VV(J)=VPMM(J)*COST-WPMM(J)*SINT
      WW(J)=WPM(J)*COST+VPM(J)*SINT
  860 A(J)=V(J)+W(J)
      GO TO 890

  865 IF(IT)880,880,870
  870 DO 875 J=1,NPANEL
      US(J)=U(J)
      VS(J)=VPMM(J)*COST-WPMM(J)*SINT
      WS(J)=WPM(J)*COST+VPM(J)*SINT
  875 AS(J)=V(J)+W(J)
      IF(IT)890,890,5
  880 DO 885 J=1,NPANEL
      UU(J)=US(J)-U(J)*SL
      VV(J)=VS(J)-(VPMM(J)*COST-WPMM(J)*SINT)*SL
      WW(J)=WS(J)-(WPM(J)*COST+VPM(J)*SINT)*SL
  885 A(J)=AS(J)-(V(J)+W(J))*SL
      IF(THKW)870,855,890
```

```
890 WRITE(NTAPEA)(A(J),J=1,NPANEL)                                    494
    WRITE(NTAPEB)(UU(J),VV(J),WW(J),J=1,NPANEL)                       499
    IF(IT)900,895,900
895 IF(THKW)35,855,900
900 CONTINUE


    GC TO 980
950 WRITE(NTAPEO,960)                                                 513
960 FORMAT(1HO,41HERROR RETURN. WING LIES OUTSIDE MACH CONE)
    GO TC 980
980 END FILE NTAPEA                                                   515
    END FILE NTAPEB
                                                                      516
    RETURN
    END
```

```
      SUBROUTINE EVAL1(NI,THKW,XC,YC,ZC,UF,VF,WF,NSTCP)                  EVAL1
C     COMPUTES AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND VELOCITY
C     COMPONENTS MATRIX
      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))     2   3   4
      XCPT=XC
      DIMENSION X(210,3,4),Y(210,3,4),Z(210,3,4),NROW(2),NPART(210)
      DIMENSIONALPHAS(210),THETA(210)
      DIMENSION XBB(50),R(50),WT(120),
      COMMON /FLOV1/KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM
      COMMON /FLOV2/X,Y,Z,NPART,ALPHAS,THETA,XBB,R,WT,T,TC,SST,CHORD
      DIMENSION T(50),TC(50),SST(210),CHORD(210)
      DIMENSION      C(4),Q(4),CPM(4),RPM(4),RPMM(4),CPMM(4),BBETAM(4)
      EPS=0.0001
      BETA=SQRT(ABS(XMACH**2-1.))                                        7
      PI=3.1415926
      CONSTA=BETA/4.0
      CONSTB=CONSTA/PI
      CONSTD=BETA/PI
      IT=1
      ISYM=1
      IF(SYM.NE.0.) ISYM=2
      JEL=1
C     INFLUENCING PANEL I
      UF=0.
      VF=0.
      WF=0.
      GO TO (1,980,3),KACE
    1 NC=NPANEL/NROW(1)
      NR=NROW(1)
      GO TO 4
    3 NC=NWING/NROW(2)
      NR=NROW(2)
    4 N=1
      NCT=NI-1
      DO 900 I=NI,NSTOP                                                  EVAL1
    5 IF(THKW)10,40,10
   10 SL=1.0
      IT=1+I-NI-NR*(N-1)
      IF(IT)30,30,25
   25 N=N+1
      THKW=1.0
      GO TO 40
   30 THKW=-1.0
      NCT=NCT+1
      GO TO 40
   35 SL=-1.0
```

```
      THKW=1.0
   40 COST=COS(THETA(I))                                                38
      SINT=SIN(THETA(I))                                                40
      TANA= TAN(ALPHAS(I))                                              42
      ABR=BETA*TANA
      IF(ABR-1.0)50,950,950
   50 IF(ABS(ABR)-EPS)80,90,85
 80   ABR=0.0
      SA=1.0
      GO TO 90
 85   SA=SIGN(1.,ABR)
 90   APM2=ABR*ABR
      NPT=NPART(I)

      U=0.
      V=0.
      W=0.
      VPM=0.
      WPM=0.
      VPMM=0.
      WPMM=0.

C     INFLUENCED PANEL J

      DO 850M=1,NPT
      BBETAM(1)=(X(I,M,2)-X(I,M,1))/(BETA*((Y(I,M,2)-Y(I,M,1))*COST
     1+(Z(I,M,2)-Z(I,M,1))*SINT))
      BBETAM(3)=(X(I,M,4)-X(I,M,3))/(BETA*((Y(I,M,4)-Y(I,M,3))*COST
     1+(Z(I,M,4)-Z(I,M,3))*SINT))
      BBETAM(2)=BBETAM(1)
      BBETAM(4)=BBETAM(3)

C     CORNER POINTS K OF INFLUENCING PANEL I

      DO 810K=1,4
      C(K)=0.0
      Q(K)=0.0
      RPM(K)=0.
      CPM(K)=0.
      RPMM(K)=0.
      CPMM(K)=0.
      BPM=BBETAM(K)
      IF(THKW)99,99,91
   91 IF(SL)92,99,93
   92 IF(K-2)810,810,99
   93 IF(K-2)99,99,810
   99 IF(ABS(BPM)-EPS)105,105,100
 100  SM=SIGN(1.0,BPM)
      GO TO 110
 105  SM=1.0
      BPM=0.0
 110  IF (ABS(ABS(BPM)-1.0)-EPS) 115,115,120
 115  BPM=SM
 120  BPM=SM*BPM
```

```
      BPM2=BPM*BPM
      CONSTC=1.+APM2*BPM2

C     CONFIGURATION SYMMETRY CONDITION

      DO 800ISIDE=1,ISYM
      SN=(-1.0)**ISIDE                                                93
      DELTAT=THETA(I)
      SINDT=SIN(DELTAT)                                               95
      COSDT=COS(DELTAT)                                               96
      XPM=XCPT-X(I,M,K)
      DELTAY=SN*(-YC)-Y(I,M,K)
      DELTAZ=ZC-Z(I,M,K)
      YPM=(DELTAY*COST+DELTAZ*SINT)*SM
      ZBR=DELTAZ*COST-DELTAY*SINT
      IF (ABS(ZBR)-EPS) 140,140,145
 140  ZBR=0.0
 145  SZ=SIGN(1.,ZBR)
      ZPM2=ZBR*ZBR
      IF (ABS(YPM)-EPS) 150,150,155
 150  YPM=0.0
 155  SY=SIGN(1.,YPM)
      XIPM=XPM/BETA
      XIPM2=XIPM*XIPM
      YPM2=YPM*YPM
      IF (THKW) 700,190,700
 190  CONTINUE
      IF (ABR) 195,499,198

C     VELOCITY COMPONENTS INDUCED BY PANELS NON-PARALLEL TO X- AXIS

 195  APM=-ABR
      ZPM=-ZBR
      GO TO 199
 198  APM=ABR
      ZPM=ZBR
 199  CONTINUE
      IF (XIPM-SQRT(YPM2+ZPM2)-EPS) 200,200,250                       113
 200  IF (BPM-1./SQRT(1.-APM2)) 205,550,550                           116
 205  TERMA=SQRT(1.-BPM2*(1.-APM2))                                   119
      TERMB=ZPM-APM*BPM*YPM
      TERMC=YPM+APM*BPM*ZPM
      TERMD=1.+APM2*BPM2
      IF (((BPM*TERMC+ABS(TERMB)*TERMA)/TERMD)-XIPM)210,210,550
 210  IF (ZPM-APM*XIPM+EPS) 225,215,215
 215  IF (YPM-BPM*XIPM*(1.-APM*TERMA)/TERMD)550,550,220
 220  P=CONSTA
      IF (ABS(ZPM-APM*XIPM)-EPS) 221,221,222
 221  P=P*SA
 222  D=CONSTA/CONSTC*(APM*BPM+TERMA)*SA
 223  S=CONSTA/CONSTC*BPM*(1.-APM*TERMA)
      GO TO 795
 225  IF (YPM-BPM*XIPM*(1.+APM*TERMA)/TERMD)550,550,230
```

```
                                                                        08/08/67
       EVAL1       -  EFN    SOURCE STATEMENT  -  IFN(S)  -

  339 FTR2=-PI
  340  IF (YPM) 370,345,370
  345  IF (TERMC) 355,350,350
  350  FTR3=FTR1
       GO TO 365
  355  IF (ZPM) 350,350,360
  360  FTR3=PI+FTR1
  365  FTR5=-APM2*BPM2*FTR3
       GO TO 415
  370  IF (TERMB) 400,375,400
  375  IF (TERMA) 380,390,385
  380  IF (YPM) 385,390,395
  385  FTR3=0.0
       GO TO 415
  390  FTR3=PI/2.0
       GO TO 415
  395  FTR3=PI*STERMB
       GO TO 415
  400  FTR3=((-XIPM*TERMF+BPM*(YPM2+ZPM2))/SQRT((YPM2+ZPM2)*TERME))             215
       IF (FTR3+1.0) 395,395,405
  405  IF (FTR3-1.0) 410,385,385
  410  FTR3=ACOS (FTR3)*STERMB                                                  220
  415  FTR3=FTR3*SIGN(1.,APM)
       FTR4=SQRT(ABS(BPM2*TERMD-1.))                                            222
       IF (YPM) 420,430,420
  420  FTR5=FTR3-FTR1*CONSTC
       IF (YPM) 430,430,422
  422  IF (TERMC) 424,430,430
  424  IF (TERMB) 426,428,430
  426  FTR5=FTR5-2.*PI*CONSTC
       GO TO 430
  428  FTR5=FTR5-PI*(2.+ABP2*BPM2)
  430  IF (YPM2+ZPM2) 435,440,435
  435  FTR6=ACOSH(XIPM/SQRT(YPM2+ZPM2))                                         236
       GO TO 445
  440  FTR6=1000.
  445  IF (TERMC) 455,450,455
  450  IF (YPM) 455,460,455
  455  FTR7=ACOSH((XIPM-APM*ZPM)/SQRT(TERMC2+TERMD*YPM2))*SQRT(TERMD)          244    245
       GO TO 465
  460  FTR7=1000.
  465  FTR8=FTR6-CONSTC*FTR7
       S=CONST8/CONSTC*(APM*BPM*FTR4*FTR2+BPM*FTR3+FTR8/APM)
       D=CONST9/CONSTC*(FTR4*FTR2-BPM*FTR6+FTR5/APM)*(-SA)
       P=CONST8*FTR1
       IF (TERMC) 795,470,795
  470  P=P*SA
       GO TO 795

C      VELOCITY COMPONENTS INDUCED BY PANELS PARALLEL TO X- AXIS

  499  CONTINUE
       ZPM=ZBR
       TERMA=(XIPM-BPM*YPM)**2
```

```
230   P=-CONSTA
      IF (TERMB) 235,240,235
235   S=-CONSTA/CONSTC*BPM*(1.+APM*TERMA)
      D=CONSTA/CONSTC*(-APM*BPM2+TERMA)*SA
      GO TO 795
240   S=-CONSTA/CONSTC*APM*TERMA*BPM
      D=CONSTA/CONSTC*TERMA*SA
      GO TO 795
250   TERMA=BPM*YPM-XIPM
      IF (ABS(TERMA)-EPS) 255,255,260
255   TERMA=0.0
260   TERMA2=TERMA*TERMA
      TERMB=APM*BPM*YPM-ZPM
      IF (ABS(TERMB)-EPS) 262,262,265
262   TERMB=0.0
      STERMB=1.0
      GO TO 270
265   STERMB=SIGN(1.0,-TERMB)
270   TERMB2=TERMB*TERMB
      TERMC=ZPM-APM*XIPM
      IF (ABS(TERMC)-EPS)275,275,280
275   TERMC=0.0
      STERMC=1.0
      GO TO 285
280   STERMC=SIGN(1.0,TERMC)
285   TERMC2=TERMC*TERMC
      TERMD=1.-APM2
      TERME=TERMA2+BPM2*TERMC2-TERMB2
      TERMF=YPM+APM*BPM*ZPM
      IF (XIPM-SQRT(YPM2+ZPM2)) 550,550,288                          153
288   IF (TERMC) 320,290,320
290   IF (TERMA) 300,305,310
300   IF (YPM) 310,305,315
305   FTR1=PI/2.
      GO TO 325
310   FTR1=0.0
      GO TO 325
315   FTR1=PI*STERMC
      GO TO 325
320   FTR1=(YPM*(TERMA-APM*TERMB)+BPM*TERMC2)/SQRT
     1((TERMC2+TERMD*YPM2)*TERME)                                    168
      IF (FTR1+1.0) 315,315,322
322   IF (FTR1-1.0) 324,310,310
324   FTR1=ACOS(FTR1)*STERMC                                         173
325   IF (BPM-1./SQRT(TERMD))335,330,330                             175
330   FTR2=ACOSH((BPM*XIPM-TERMF)/SQRT(TERME))                       178
      GO TO 340
335   FTR2=(BPM*XIPM-TERMF)/SQRT(TERME)                              181
      IF (FTR2+1.0) 339,339,336
336   IF (FTR2-1.0) 337,338,338
337   FTR2=-ACOS(FTR2)                                               186
      GO TO 340
338   FTR2=0.0
      GO TO 340
```

```
                                                                          08/08/67
        EVAL1        -  EFN    SOURCE STATEMENT  -  IFN(S)  -

      TERMB=YPM2+ZPM2
      TERMC=SQRT(TERMB)                                                   253
      TERMD=BPM2-1.0
      TERME=-TERMD
      IF (BPM2-1.0) 500,560,560
  500 IF (XIPM-TERMC)520,535,510
  510 D=CONSTB*(SQRT(TERME)*ACOS((BPM*XIPM-YPM)/SQRT
     1(TERMA+TERMD*ZPM2))+BPM*ACCSH(XIPM/TERMC)+
     2YPM/TERMB*SQRT(XIPM2-TERMB))                                        258    259    260    261
      GO TO 575
  520 IF (YPM) 550,550,530
  530 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,537,535               266
  535 IF (YPM-BPM*XIPM)550,537,540
  537 CT=.5
      GO TO 545
  540 CT=1.0
  545 D=CONSTA*SQRT(TERME)*CT                                             274
      S=CONSTA*BPM*SZ*CT
      P=CONSTA*SZ*CT
      GO TO 795
  550 D=0.0
      S=0.0
      P=0.0
      GO TO 795
  560 IF (XIPM-TERMC) 550,550,570
  570 D=CONSTB*(-SQRT(TERMD)*ACCSH((BPM*XIPM-YPM)
     1/SQRT(TERMA+TERMD*ZPM2))+BPM*ACCSH(XIPM/TERMC)
     2+YPM/TERMB*SQRT(XIPM2-TERMB))                                       281    282    283
  575 IF (ZPM) 580,585,580
  580 FACTOR=     (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*                       287
     1(TERMA+TERMC*ZPM2))
      IF (FACTOR+1.) 600,600,582
  582 IF (FACTOR-1.) 584,590,590
  584 FACTOR=ACOS(FACTOR)*SZ                                              292
      GO TO 625
  585 IF (YPM) 590,597,595
  590 FACTOR=0.0
      GO TO 625
  595 IF (XIPM-BPM*YPM) 590,597,600
  597 FACTOR=PI/2.
      GO TO 625
  600 FACTOR=PI*SZ
  625 S=CONSTB*(BPM*FACTOR-ZPM/TERMB*SQRT(XIPM2-TERMB))                   304
      P=CONSTB*FACTOR
      GO TO 795
  700 CONTINUE

C     VELOCITY COMPONENTS INDUCED BY WING SOURCES

      ZPM=ZBR
      TERMAA=XIPM-BPM*YPM
      TERMA=(XIPM-BPM*YPM)**2
      TERMB=YPM2+ZPM2
```

```
      TERMC=SQRT(TERMB)                                                  307
      TERMD=BPM2-1.0
      TERME=-TERMC
      IF (XIPM-TERMC) 704,704,708
 704  IF(TERMD)753,550,550
 708  FTR1=ACOSH(XIPM/TERMC)
      IF (TERMD) 712,716,720
 712  FTR2=ACOS((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERME)       314   315   316
      GO TO 724
 716  FTR2=SQRT(XIPM2-TERMB)/(XIPM-YPM)                                  319
      GO TO 724
 720  FTR2=ACOSH((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERMD)      322   323
 724  IF (ZPM) 728,732,728
 728  FTR3=     (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*(TERMA+TERMD*ZPM2))     327
      IF (FTR3+1.0) 748,748,730
 730  IF (FTR3-1.0) 731,736,736
 731  FTR3=ACOS(FTR3)*SZ                                                 332
      GO TO 750
 732  IF (YPM) 736,736,740
 736  FTR3=0.
      GO TO 750
 740  IF (XIPM-BPM*YPM) 736,744,748
 744  FTR3=PI/2.
      GO TO 750
 748  FTR3=PI*SZ
 750  IF(THKW)751,190,752
 751  P= CONSTD*(TERMAA*FTR2+YPM*FTR1-ZPM*FTR3)/CHORD(NCT)
      S=-CONSTD*(TERMAA*(FTR1-BPM*FTR2)+BPM*ZPM*FTR3
     1-SQRT(XIPM2-TERMB))/CHORD(NCT)                                     347
      D=-CONSTD*(ZPM*(BPM*FTR1-TERMD*FTR2)-TERMAA*FTR3)
     1/CHORD(NCT)
      GO TO 795
 752  P=FTR2/PI
      S=-(FTR1-BPM*FTR2)/PI
      D=FTR3/PI
      GO TO 795
 753  IF(YPM)550,550,756
 756  IF (XIPM-TERMC) 760,764,708
 760  IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM))) 550,768,764               358
 764  IF (YPM-BPM*XIPM) 550,768,772
 768  CT=.5
      GO TO 776
 772  CT=1.
 776  IF(THKW)777,190,778
 777  P=(TERMAA/SQRT(TERME)-ZPM*CT*SZ)*BETA/CHORD(NCT)                   368
      S=BPM*P
      D=BETA*(-ZPM*SQRT(TERME)+TERMAA*CT*SZ)/CHORD(NCT)                  370
      GO TO 795
 778  P=CT/SQRT(TERME)                                                   374
      S=BPM*P
      D=CT*SZ

 795  CONTINUE
```

```
      IF (ISIDE-1) 796,796,797
796   SIM=1.
      GO TO 799
797   IF (THKW) 796,798,796
798   SIM=SYM
799   CONTINUE

      Q(K)=Q(K)-P*SM*SIM
      RPM(K)=RPM(K)+S*SIM
      CPM(K)=CPM(K)+D*SM*SIM
      RPMM(K)=RPMM(K)-S*SN*SIM
      CPMM(K)=CPMM(K)-D*SM*SN*SIM
800   CONTINUE
810   CONTINUE

      U=U+(Q(1)-Q(2)-Q(3)+Q(4))/BETA
      VPM=VPM+(RPM(1)-RPM(2)-RPM(3)+RPM(4))
      WPM=WPM+(CPM(1)-CPM(2)-CPM(3)+CPM(4))
      VPMM=VPMM+(RPMM(1)-RPMM(2)-RPMM(3)+RPMM(4))
      WPMM=WPMM+(CPMM(1)-CPMM(2)-CPMM(3)+CPMM(4))
850   CONTINUE

      IF(THKW)865,855,865
855   CONTINUE
      UU=U
      VV=VPMM*COST-WPMM*SINT
      WW=WPM*COST+VPM*SINT
      GO TO 890

865   IF(IT)880,880,870
870   CONTINUE
      US=U
      VS=VPMM*COST-WPMM*SINT
      WS=WPM*COST+VPM*SINT
      IF(IT)890,890,5
880   CONTINUE
      UU=US-U*SL
      VV=VS-(VPMM*COST-WPMM*SINT)*SL
      WW=WS-(WPM*COST+VPM*SINT)*SL
      IF(THKW)870,855,890
890   CONTINUE
      IF(THKW.NE.0.) GO TO 8991
      UF=UF+UU*SST(I)
      VF=VF+VV*SST(I)
      WF=WF+WW*SST(I)
      GO TO 8992
8991  UF=UF+UU*WT(JEL)
      VF=VF+VV*WT(JEL)
      WF=WF+WW*WT(JEL)
      JEL=JEL+1
8992  CONTINUE
      IF(IT) 899,895,899
895   IF(THKW) 35,855,899
```

```
899 CONTINUE
900 CONTINUE


    GO TO 980
950 WRITE(NTAPEO,960)                                                    429
960 FORMAT(1H0,41HERROR RETURN. WING LIES OUTSIDE MACH CONE)


980 CONTINUE
    RETURN
    END
```

```
       FASNCS
     ASSEMBLED TEXT.

            $TEXT  FASNCS
                                                                          FASN0001
                                    *ASNCS   IBJOB LIBRARY / ASIN AND ACOS                                      ASIN0010
                                    *        ADAPTED FROM SHARE SUBROUTINE B1 CL ASC1                           ASIN0020
                                    *        CONVERTED TO IBMAP / 1-9-63 / E. WRIGHT - TRANSPORT - BOEING       ASIN0030
                                    *        METHODS MEMO 7090-49                                               ASIN0040
                                    *                                                                           ASIN0050
                                    *                                                                         * ASIN0060
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       ASIN0070
                                    *                                                                           ASIN0080
                                    *      A=ASIN (B)                                                           ASIN0090
                                    *      A=ACOS (B)                                                           ASIN0100
                                    *                                                                         * ASIN0110
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       ASIN0120
                                                                                                                ASIN0130
                                                                                                                ASIN0140
                                             ENTRY   ASIN                                                       ASIN0150
                                             ENTRY   ACOS                                                       ASIN0160
                                                                                                                ASIN0170
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     ASIN0180

BINARY CARD (NOT PUNCHED)                                                                                       ASIN0190
      00000  0600 00 0 04013  10011  ASIN    STZ     BEANIE          ASIN ENTRY                                 ASIN0200
      00001  0020 00 C 00003  10001          TRA     PETUN                                                      ASIN0210
      00002  4625 00 0 04013  10011  ACOS    STL     BEANIE          ACOS ENTRY                                 ASIN0220
      00003  1 00000 C 00006  10001  PETUN   SAVEN
      00004  0774 00 4 00000  10000
      00005  0020 00 4 00001  10000
      00006  0634 00 4 07000  10011
      00007  0634 00 4 00004  10001                                                                             ASIN0230
      00010  0634 00 4 00121  10001          SXA     LK.DR,4                                                    ASIN0240
      00011  0500 60 4 00003  10000          CLA*    3,4             GET ARGUMENT X                             ASIN0250
      00012  0601 00 C 04015  10011  ASIN3   STO     X               STORE X                                    ASIN0260
      00013  0500 00 C 00123  10001          CLA     =1.0                                                       ASIN0270
      00014  0306 00 0 04015  10011          FSM     X               1 - ABS X                                  ASIN0280
      00015  0601 00 0 04014  10011          STO     TEMP                                                       ASIN0290
      00016  4120 00 0 00073  10001          TMI     ERROR                                                      ASIN0300
      00017  0500 00 C 04015  10011          CLA     X                                                          ASIN0310
      00020  0520 00 C 04013  10011          ZET     BEANIE          WAS ENTRY ASIN OR ACOS                     ASIN0320
      00021  0020 00 0 00034  10001          TRA     ACOS1                                                      ASIN0330
      00022  0560 00 0 00124  10001  ASIN6   LDQ     =O171400000000  2(-8) = .00390625

BINARY CARD (NOT PUNCHED)                                                                                       ASIN0340
      00023  0760 00 C 00003  10000          SSP                                                                ASIN0350
      00024  0040 00 0 00027  10001          TLQ     ASIN10                                                     ASIN0360
      00025  0500 00 0 04015  10011          CLA     X               X EQUALS ARCSIN X                          ASIN0370
                      00026                  RETURN  PETUN                                                      ASIN0380
      00027  0500 00 0 04015  10011  ASIN10  CLA     X                                                          ASIN0390
      00030  0560 00 0 00125  10001          LDQ     =1.57079633     PI/2                                       ASIN0400
      00031  0765 00 C 00000  10000          LRS     0               + OR - PI/2                                ASIN0410
      00032  0760 00 C 00002  10000          CHS                     -X                                         ASIN0420
      00033  0020 00 0 00037  10001          TRA     ASIN23                                                     ASIN0430
      00034  0560 00 C 00126  10001  ACOS1   LDQ     =0              ZERO                                       ASIN0440
      00035  0120 00 0 00037  10001          TPL     ASIN23
```

```
         FASNCS
      ASSEMBLED TEXT.

        00036  0560 00 0 00127  10001         LDQ    =3.14159265     PI                                ASIN0450
        00037  4600 00 0 04012  10011  ASIN23 STQ    A               A                                 ASIN0460
        00040  0601 00 0 04013  10011         STO    BEANIE          PLUS OR MINUS (8)                 ASIN0470
        00041  000000000000     00010         CALL   SQRT(TEMP)      SQUARE ROOT (1-ABS X)             ASIN0480
        00041  0074 00 4 06000  10011
        00042  1 00001 0 01003  10011
        00043  0 00121 0 00072  10100
        00044  0 00000 0 04014  10011

BINARY CARD (NOT PUNCHED)
        00045  0560 00 0 04013  10011         LDQ    BEANIE                                            ASIN0490
        00046  0763 00 0 00000  10000         LLS    0                                                 ASIN0500
        00047  0601 00 0 04013  10011         STO    BEANIE          PLUS OR MINUS 8 SQRT (1-ABS X)    ASIN0510
        00050  0560 00 0 04015  10011         LDQ    X               X                                 ASIN0520
        00051  0763 00 0 00010  10000         LLS    8               GET CHARACTERISIC                 ASIN0530
        00052  0734 00 4 00000  10000         PAX    0,4             PLACE IN IR4                      ASIN0540
        00053  4754 00 0 00000  10000         ZAC                                                      ASIN0550
        00054  0765 00 4 00200  10000         LRS    128,4           SHIFT(128 - POWER)MODULO 256      ASIN0560
        00055  4600 00 0 04015  10011         STQ    X               X IN FIXED POINT                  ASIN0570
        00056  0500 00 0 00101  10001         CLA    A8              A8                                ASIN0580
        00057  0774 00 4 00010  10000         AXT    8,4                                               ASIN0590
        00060  0131 00 0 00000  10000  ASIN41 XCA                                                      ASIN0600
        00061  4200 00 0 04015  10011         MPR    X               TIMES X                           ASIN0610
        00062  0400 00 4 00112  10001         ADD    A0+1,4          PLUS A(I)                         ASIN0620
        00063  2 00001 4 00060  10001         TIX    ASIN41,4,1                                        ASIN0630
        00064  0765 00 0 00010  10000         LRS    8                                                 ASIN0640
        00065  0760 00 0 00010  10000         RND                                                      ASIN0650
        00066  0400 00 0 00130  10001         ADD    =129B8          FLOAT                             ASIN0660
        00067  0131 00 0 00000  10000         XCA                                                      ASIN0670

BINARY CARD (NOT PUNCHED)
        00070  0260 00 0 04013  10011         FMP    BEANIE          TIMES 8 SQRT (1-ABS X)            ASIN0680
        00071  0300 00 0 04012  10011         FAD    A               PLUS A                            ASIN0690
                         00072                RETURN PETUN           EXIT                              ASIN0700
                                       *                                                               ASIN0710
                                       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0720
                                       *                                                               ASIN0730
        00073  000000000000     00010  ERROR  CALL   .FXEM.(BAIL)    CALL ERROR EXECUTION MONITOR      ASIN0740
        00073  0074 00 4 05000  10011
        00074  1 00001 0 01003  10011
        00075  0 00121 0 00136  10100
        00076  0 00000 0 00112  10001
        00077  0500 00 0 00123  10001         CLA    =1.0            SET ARGUMENT TO ONE               ASIN0750
        00100  0020 00 0 00012  10001         TRA    ASIN3                                             ASIN0760
                                       *                                                               ASIN0770
                                       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0780
                                       *                                                               ASIN0790
        00101  000076257520     10000  A8     OCT    76257520        A8                                ASIN0800
        00102  400507371460     10000         DEC    -85848880       A7                                ASIN0810
        00103  0 01472 4 40066  10000         PZE    16438,4,826     A6                                ASIN0820
        00104  402641537160     10000         DEC    -377929328      A5                                ASIN0830
        00105  004173275020     10000         OCT    4173275020      A4                                ASIN0840
        00106  406374356540     10000         DEC    -871488864      A3                                ASIN0850
        00107  013313544557     10000         OCT    13313544557     A2                                ASIN0860
```

```
           FASNCS
        ASSEMBLED TEXT.

     00110  433360112561     10000          OCT     -033360112561  A1                              ASIN0870
     00111  311037552421     10000   A0     OCT     311037552421   A0                              ASIN0880

BINARY CARD (NOT PUNCHED)
     00112  000000000101     10000   BAIL   DEC     65                                             ASIN0881
     00113  0 00002 C 00115  10001          PZE     MESA,,2                                        ASIN0882
     00114  0 00002 C 00117  10001          PZE     MESB,,2                                        ASIN0883
     00115  002151276027     10000   MESA   BCI     2,0ARG GT ONE                                  ASIN0884
     00116  636046452560     10000
     00117  002151276013     10000   MESB   BCI     2,0ARG = 1                                     ASIN0885
     00120  600160606060     10000
                                     *                                                             ASIN0890
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0900
                                     *                                                             ASIN0910
     00121  00000000C000     10000   LK.DR  LDIR                                                   ASIN0920
     00122  262162452362     10000
     00123  20140000C000     10000          LORG                                                   ASIN0930
     00124  17140000C000     10000
     00125  201622077325     10000
     00126  000C00000000     10000
     00127  202622077325     10000
     00130  201C0000C000     10000
                                     *                                                             ASIN0940
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN0950
                                     *                                                             ASIN0960
                                      ERASE CONTRL  ER                                             ASIN0970
     00131  10000000C131     00001          USE     ER                                             ASIN0980
     00131  20000000C012     00001          BSS     10           RESERVE FOR SQRT                  ASIN0990
     00143  20000000C001     00001   A      BSS     1                                              ASIN1000
     00144  20000000C001     00001   BEANIE BSS     1                                              ASIN1010

BINARY CARD (NOT PUNCHED)
     00145  200000000001     00001   TEMP   BSS     1                                              ASIN1020
     00146  20000000C001     00001   X      BSS     1                                              ASIN1030
     00147  20000000C006     00001          BSS     6                                              ASIN1040
                                     *                                                             ASIN1050
                                     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASIN1060
                                                                                                   ASIN1070
                   00000  01111             END                                                    ASIN1080
```

```
      FASNCS
    CONTROL DICTIONARY

          $CDICT FASNCS                                                          FASN0002

BINARY CARD (NOT PUNCHED)
          000155000000              PREFACE     START=0,LENGTH=109,TYPE=7094,CMPLX=5
          000004000005
          262162452362       FASNCS DECK        LOC=0,LENGTH=109
          000155000000
          216231456060       ASIN   REAL        LOC=0,LENGTH=0
          000000000000
          212346626060       ACOS   REAL        LOC=2,LENGTH=0
          000000000002
          255121622560       ERASE  REAL        LOC=131,LENGTH=20
          000024000131
          332667254433       .FXEM. VIRTUAL     SECT. 5,CALL
          200000100000
          625051636060       SQRT   VIRTUAL     SECT. 6,CALL
          200000100000
          627062434623       SYSLOC VIRTUAL     SECT. 7
          200000000000

          $DKEND FASNCS                                                          FASN0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
        FATN1                                            07/31/65
       ASSEMBLED TEXT.

             *TEXT   FATN1                                                                 FATN0001

                                         *     SINGLE-VALUED ARCTANGENT FUNCTION / ATN1               ATN10010
                                         * SHARE SUBROUTINE B1 GM ATN1                                ATN10040
                                         * METHODS MEMO 7090-45                                       ATN10050
                                         *                                                            ATN10060
                                         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10070
                                         *                                                            ATN10080
                                         *     A=ATN1 (Y,X)                                           ATN10090
                                         *                                                            ATN10100
                                         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10110
                                                                                                      ATN10120
                                               ENTRY   ATN1                                           ATN10130
                                                                                                      ATN10140

BINARY CARD (NOT PUNCHED)
       00000  000000000000        10000   LNKG  LDIR                                                  ATN10150
       00001  262163450160        10000
                                                                                                      ATN10160
                                         * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN10170
       00002  1 00000 0 00010     10001   STRT  SAVEN   2,I                                           ATN10180
       00003  0774 00 2 00000     10000
       00004  0774 00 4 00000     10000
       00005  0441 00 0 00007     10001
       00006  0020 00 4 00001     10000
       00007  0 00000 0 00000     10000
       00010  0604 00 0 00007     10001
       00011  0634 00 4 04000     10011
       00012  0634 00 4 00004     10001
       00013  0634 00 2 00003     10001
       00014  0634 00 4 00000     10001         SXA     LNKG,4                                        ATN10190
       00015  0500 60 4 00004     10000         CLA*    4,4                                           ATN10200
       00016  0131 00 0 00000     10000         XCA                                                   ATN10210
       00017  0500 60 4 00003     10000         CLA*    3,4                                           ATN10220
       00020  0601 00 0 03023     10011         STO     COMMON                                        ATN10230
       00021  0020 00 0 00023     10001         TRA     BGN                                           ATN10240
       00022  0020 00 0 00002     10001   ATN1  TRA     STRT                                          ATN10250

BINARY CARD (NOT PUNCHED)
       00023  4600 00 0 03022     10011   BGN   STQ     COMMON-1            STORE X                   ATN10260
       00024  0100 00 0 02023     10011         TZE     ATN1+19                                       ATN10270
       00025  0771 00 0 00033     10000         ARS     27                  CHAR. IN ACC EQUALS P     ATN10280
       00026  0760 00 0 00003     10000         SSP                                                   ATN10290
       00027  0601 00 0 03020     10011         STO     COMMON-3            CHAR. OF Y                ATN10300
       00030  0500 00 0 03022     10011         CLA     COMMON-1                                      ATN10310
       00031  0100 00 0 02025     10011         TZE     ATN1+21                                       ATN10320
       00032  0771 00 0 00033     10000         ARS     27                  CHAR. OF X IN ACC EQUALS Q ATN10330
       00033  0760 00 0 00003     10000         SSP                                                   ATN10340
       00034  0402 00 0 03020     10011         SUB     COMMON-3            Q-P                       ATN10350
       00035  0402 00 0 02147     10011         SUB     ATN1+103            Q-P-1                     ATN10360
       00036  0601 00 0 03017     10011         STO     COMMON-4                                      ATN10370
       00037  0760 00 0 00003     10000         SSP                         ABS VALUE OF ABOVE        ATN10380
       00040  0402 00 0 02164     10011         SUB     ATN1+116            I Q-P-1 I-128             ATN10390
       00041  4120 00 0 02027     10011         TMI     ATN1+23             XFER IF IN RANGE          ATN10400
```

```
        FATN1
      ASSEMBLED TEXT.                                                   07/31/65

      00042  0500 00 0 03020  10011      CLA    COMMON-3          Q-P                               ATN10410
      00043  0402 00 0 02164  10011      SUB    ATN1+115          Q-P-128                           ATN10420
      00044  0120 00 0 02025  10011      TPL    ATN1+21           XFER IF Y/X WOULD O-LOW           ATN10430
      00045  4754 00 0 00000  10000      PXD    0,0               ZERO ACC, IF Y/X IS ZERO          ATN10440

BINARY CARD (NOT PUNCHED)
      00046  0020 00 0 02122  10011      TRA    ATN1+82           SET Z EQUALS ZERO                 ATN10450
      00047  0500 00 0 02146  10011      CLA    ATN1+102          PI/2, IF Y/X IS INF.,             ATN10450
      00050  0020 00 0 02122  10011      TRA    ATN1+82           SET Z EQUALS PI/2                 ATN10470
      00051  0500 00 0 03022  10011      CLA    COMMON-1          X                                 ATN10480
      00052  0300 00 0 02145  10011      FAD    ATN1+101          NORMALIZE                         ATN10490
      00053  0601 00 0 03021  10011      STO    COMMON-2                                            ATN10500
      00054  0500 00 0 03023  10011      CLA    COMMON            Y                                 ATN10510
      00055  0241 00 0 03021  10011      FDP    COMMON-2          Y/X                               ATN10520
      00056  4600 00 0 03021  10011      STQ    COMMON-2                                            ATN10530
      00057  0500 00 0 03021  10011      CLA    COMMON-2                                            ATN10540
      00060  0760 00 0 00003  10000      SSP                      ABS VAL OF Y/X                    ATN10550
      00061  0601 00 0 03021  10011      STO    COMMON-2          EQUALS W                          ATN10560
      00062  0760 00 0 00000  10000      CLM                                                        ATN10570
      00063  0601 00 0 03013  10011      STO    COMMON-8          SET SW1 TO ZERO                   ATN10580
      00064  0500 00 0 03017  10011      CLA    COMMON-4          Q-P-1                             ATN10590
      00065  0760 00 0 00003  10000      SSP                      ABS VALUE                         ATN10600
      00066  0402 00 0 02166  10011      SUB    ATN1+118          I Q-P-1 I-10                      ATN10610
      00067  4120 00 0 02053  10011      TMI    ATN1+43           XFER IF W BETWN 2**-10+2**10      ATN10620
      00070  0500 00 0 02147  10011      CLA    ATN1+103          1                                 ATN10630

BINARY CARD (NOT PUNCHED)
      00071  0601 00 0 03013  10011      STO    COMMON-8          RESET SW1 TO 1                    ATN10640
      00072  0500 00 0 03017  10011      CLA    COMMON-4          Q-P-1                             ATN10650
      00073  0120 00 0 02064  10011      TPL    ATN1+52           XFER IF W GTR THAN 2**10          ATN10660
      00074  0020 00 0 02056  10011      TRA    ATN1+46           XFER IF W LS THAN 2**-10          ATN10670
      00075  0500 00 0 03021  10011      CLA    COMMON-2          W                                 ATN10680
      00076  0302 00 0 02150  10011      FSB    ATN1+104          W-1                               ATN10690
      00077  0120 00 0 02064  10011      TPL    ATN1+52           XFER IF W GTR THAN 1              ATN10700
      00100  0500 00 0 02146  10011      CLA    ATN1+102          PI/2                              ATN10710
      00101  0601 00 0 03014  10011      STO    COMMON-7          SET SW2 TO PI/2                   ATN10720
      00102  0500 00 0 02150  10011      CLA    ATN1+104          1.   IF W LS THAN 1, USE 1/W,     ATN10730
      00103  0241 00 0 03021  10011      FDP    COMMON-2          1/W  + SUBT VALUE FROM PI/2       ATN10740
      00104  4600 00 0 03021  10011      STQ    COMMON-2                                            ATN10750
      00105  0020 00 0 02066  10011      TRA    ATN1+54           XFER TO POLYNOM                   ATN10760
      00106  4754 00 0 00000  10000      PXD    0,0               ZERO ACC                          ATN10770
      00107  0601 00 0 03014  10011      STO    COMMON-7          SET SW2 TO ZERO                   ATN10780
      00110  0500 00 0 03013  10011      CLA    COMMON-8          SW1                               ATN10790
      00111  4100 00 0 02117  10011      TNZ    ATN1+79           IF NOT ZERO, BYPASS POLYNOM.      ATN10800
      00112  0500 00 0 03021  10011      CLA    COMMON-2          W                                 ATN10810
      00113  0300 00 0 02150  10011      FAD    ATN1+104          W+1                               ATN10820

BINARY CARD (NOT PUNCHED)
      00114  0601 00 0 03017  10011      STO    COMMON-4                                            ATN10830
      00115  0302 00 0 02151  10011      FSB    ATN1+105          W-1                               ATN10840
      00116  0100 00 0 02115  10011      TZE    ATN1+77           IF ZERO, Z EQUALS PI/4            ATN10850
      00117  0241 00 0 03017  10011      FDP    COMMON-4          W-1/W+1 EQUALS R                  ATN10860
      00120  4600 00 0 03020  10011      STQ    COMMON-3                                            ATN10870
      00121  0260 00 0 03020  10011      FMP    COMMON-3          R*R                               ATN10880
```

```
 00122  0601 00 0 03017  10011        STO     COMMON-4                                  ATN10890
 00123  0761 00 0 00000  10000        NOP                                               ATN10900
 00124  0534 00 2 02152  10011        LXA     ATN1+106,2       SET IRB TO 7             ATN10910
 00125  0500 00 0 02153  10011        CLA     ATN1+107         C15                      ATN10920
 00126  0601 00 0 03015  10011        STO     COMMON-6         POLYNOM ERASEABLE E      ATN10930
 00127  0560 00 0 03015  10011        LDQ     COMMON-6                                  ATN10940
 00130  0260 00 0 03017  10011        FMP     COMMON-4         E*R*R                    ATN10950
 00131  0300 00 2 02163  10011        FAD     ATN1+115,2       PLUS CI                  ATN10960
 00132  2 00001 2 02104  10011        TIX     ATN1+68,2,1      LOOP                     ATN10970
 00133  0761 00 0 00000  10000        NOP                                               ATN10980
 00134  0601 00 0 03016  10011        STO     COMMON-5         VALUE OF POLYNOM         ATN10990
 00135  0560 00 0 03016  10011        LDQ     COMMON-5         EQUALS T                 ATN11000
 00136  0260 00 0 03020  10011        FMP     COMMON-3         T*R                      ATN11010

BINARY CARD (NOT PUNCHED)
 00137  0300 00 0 02163  10011        FAD     ATN1+115         T*R+PI/4                 ATN11020
 00140  0601 00 0 03021  10011        STO     COMMON-2         EQUALS Z                 ATN11030
 00141  0500 00 0 03014  10011        CLA     COMMON-7         TEST SW2   EQU. 0 OR PI/2 ATN11040
 00142  0100 00 0 02123  10011        TZE     ATN1+83          XFER IF ZERO             ATN11050
 00143  0302 00 0 03021  10011        FSB     COMMON-2         PI/2-Z                   ATN11060
 00144  0601 00 0 03021  10011        STO     COMMON-2         PORPER 1ST QUAD Z NOW IN CMMN+2 ATN11070
 00145  0560 00 0 03023  10011        LDQ     COMMON           Y   INST 83-100 ADJUST Z TO ATN11080
 00146  0260 00 0 03022  10011        FMP     COMMON-1         Y*X         PROPER QUAD. ATN11090
 00147  0120 00 0 02136  10011        TPL     ATN1+94          XFER IF SAME SIGN        ATN11100
 00150  0500 00 0 03022  10011        CLA     COMMON-1         X                        ATN11110
 00151  0120 00 0 02132  10011        TPL     ATN1+90          XFER IF X+,Y-4TH QUAD.   ATN11120
 00152  0500 00 0 02165  10011        CLA     ATN1+117         PI      X-,Y+2ND QUAD.   ATN11130
 00153  0020 00 0 02134  10011        TRA     ATN1+92                                   ATN11140
 00154  0500 00 0 02165  10011        CLA     ATN1+117         PI     4TH QUAD          ATN11150
 00155  0300 00 0 02165  10011        FAD     ATN1+117         2*PI                     ATN11160
 00156  0302 00 0 03021  10011        FSB     COMMON-2         -Z                       ATN11170
                  00157               RETURN  STRT                                      ATN11180
 00160  0500 00 0 03022  10011        CLA     COMMON-1         X    SAME SIGN           ATN11190
 00161  0120 00 0 02143  10011        TPL     ATN1+99          XFER IF X+,Y+,1ST QUAD   ATN11200

BINARY CARD (NOT PUNCHED)
 00162  0500 00 0 02165  10011        CLA     ATN1+117         PI      X-,Y-,3RD QUAD   ATN11210
 00163  0300 00 0 03021  10011        FAD     COMMON-2         +Z                       ATN11220
                  00164               RETURN  STRT                                      ATN11230
 00165  0500 00 0 03021  10011        CLA     COMMON-2         Z                        ATN11240
                  00166               RETURN  STRT                                      ATN11250
                                *                                                       ATN11260
                                * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11270
                                *                                                       ATN11280
 00167  0000 00 0 00000  10000        HTR     0                FIRST PROGRAM CONSTANT   ATN11290
 00170  201622077325     10000        DEC     1.57079633,1,1.,2.,7,-.004054058,.0218612288 ATN11300
 00171  000000000001     10000
 00172  201400000000     10000
 00173  202400000000     10000
 00174  000000000007     10000
 00175  571411537472     10000
 00176  173546131217     10000
 00177  574712007736     10000        DEC     -.0559098861,.096420044,-.139085335,.199465360 ATN11310
 00200  175612737372     10000
```

```
        FATN1                                                 07/31/65
      ASSEMBLED TEXT.

        00201  576434661427   10000
        00202  176630402456   10000
        00203  577525230337   10000          DEC      -.333298561,.999999333,.78539816,128,3.14159265,10      ATN11320
        00204  200777777646   10000

BINARY CARD (NOT PUNCHED)
        00205  200622077325   10000
        00206  000000000200   10000
        00207  202622077325   10000
        00210  000000000012   10000
                                      *                                                                        ATN11330
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11340
                                      *                                                                        ATN11350
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11360
                                      *                                                                        ATN11370
                                      ERASE   CONTRL  POOL                                                     ATN11380
        00211  100000000211   00001           USE     POOL                                                     ATN11390
        00211  200000000023   00001    TEMP   BSS     19                                                       ATN11400
        00234  200000000001   00001   COMMON  BSS     1                                                        ATN11410
                                      *                                                                        ATN11420
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ATN11430
                                                                                                               ATN11440
                      00000   01111           END                                                              ATN11450
```

```
          FATN1                                                    07/31/65
        CONTROL DICTIONARY

              $CDICT FATN1                                                       FATN0002

BINARY CARD (NOT PUNCHED)
          000235000000             PREFACE     START=0,LENGTH=157,TYPE=7094,CMPLX=5
          000004000005
          262163450160     FATN1   DECK        LOC=0,LENGTH=157
          000235000000
          216345016060     ATN1    REAL        LOC=22,LENGTH=0
          000000000022
          255121622560     ERASE   REAL        LOC=211,LENGTH=20
          000024000211
          627062434623     SYSLOC  VIRTUAL     SECT. 4
          200000000000

              $DKEND FATN1                                                       FATN0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
      SUBROUTINE FCALC
C     CONTROL SUBROUTINE FOR CALCULATION OF FORCES, MOMENTS
C     AND PRESSURES
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     A               RWT(20),ALPHA(210),ALPHAB(210),ALPHAS(210),ALPHAT(1
     A               10),ALPHAW(110),ALPHAX(110),AWS(110)
     A               ,ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW,ARAS,AT,AAA
      COMMON /BVAR/  B(210)
     B               ,BBCL,BBCD,BBCM
      COMMON /CVAR/  C(210),CHORD(210),CL(210),CPBB(55,10),CPNN(55,10,2)
     C               ,CLS(210)
     C               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDM
      COMMON /DVAR/  D(210),DZDXB(55)
     D               ,DADEG,DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     N               ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NBODYS,NTHETA,NT
     N               HETS,NXLE,NRG,NPOLAR,NCLX
      COMMON /PVAR/  POLAR
      COMMON /RVAR/  R(55),RN(55,2)
     R               ,RATIOX,RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON /TVAR/  TITLE(12),THETA(210),THETAB(10),THETAS(9),THETAN(10
     T               ,2),TCL(10),TCD(10),TCM(10)
     T               ,THICK,TWIST
      COMMON /UVAR/  UBWT(100),UWBT(110),UWWT(110),UNCL(210,2)
      COMMON /VVAR/  VBWT(100),VWBT(110),VWWT(110),VNCL(210,2)
     V               ,VOUT
      COMMON /WVAR/  WBWT(100),WWBT(110),WWWT(110),WNCL(210,2)
      COMMON /XVAR/  XBAR(210),XC(210),XNI(2),XNXN(2),XNTN(2),XB(55),XN(
     X               55,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(10),XCD
     X               (10)
     X               ,XNACEL,XP,XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON /ZVAR/  ZBAR(210),ZC(210),ZNI(2),ZDELTA(55),ZDN(55,2)
     Z               ,ZP,ZA
      DIMENSION ALPHAL(110),ALPHAU(110)
      DIMENSION CHORDL(20),CLBB(100),CLWW(110),CPB(100),CPL(110),CPU(110
     C         ),CSHAPE(110)
      DIMENSION U(110),UBB(100),UBN(100),UBW(100),UWB(110),UWN(110),UWW(
     U         110)
      DIMENSION V(110),VBB(100),VBN(100),VBW(100),VWB(110),VWN(110),VWW(
     V         110)
```

```
      DIMENSION W(110),WBB(100),WBN(100),WBW(100),WWB(110),WWN(110),WWW(
     W          110),WSCLL(20),WSCLU(20),WSCDL(20),WSCDU(20),WSCL(20),WS
     W          CD(20)
C     KONFIG = 1     WING ALONE CONFIGURATION
C     KONFIG = 2     BODY ALONE CONFIGURATION
C     KONFIG = 3     WING-BODY CONFIGURATION

      BETA=SQRT(ABS(XMACH*XMACH-1.))                                       2
      BETA2=BETA*BETA
      BETAM=2.4/1.414214*XMACH**4/BETA**.75                                3
      RATIOD=.75

C     ANY CONFIGURATION
      IF (KONFIG-2) 105, 90,95

C     BODY-ALONE CONFIGURATION
   90 CONTINUE
      CALL FSF(10,NTAPEC,IRR)                                              6
      END FILE NTAPEC                                                      7
      REWIND NTAPEC                                                        8
      GO TO 550
C     WING-BODY CONFIGURATION
   95 CONTINUE
      DO 100 J=1,NBODY
      UBN(J)=0.
      VBN(J)=0.
      WBN(J)=0.
  100 CONTINUE

C     WING ALONE OR WING-BODY CONFIGURATION
  105 CONTINUE
      DO 110 J=1,NWING
      UWB(J)=0.
      VWB(J)=0.
      WWB(J)=0.
      UWN(J)=0.
      VWN(J)=0.
      WWN(J)=0.
  110 CONTINUE
      XNNCLT=0.
      XNNCDT=0.
      XNNCMT=0.


  160 CONTINUE

      IF (IPOLAR .NE. 0) GO TO 210

      GO TO (170,210,260),KASE

C     CALCULATE WING ALPHA, GIVEN WING CL      KASE = 1
  170 CONTINUE

      IF (KONFIG-2) 180,190,190
```

```
C     WING ALONE CONFIGURATION
  180 CONTINUE
      CALL CAMBW (NWING,NTAPEE,A,CL,ALPHAW)                                41
      REWIND NTAPEE                                                        42
      GO TO 360

C     WING-BODY CONFIGURATION
  190 CONTINUE
      CALL FSF (1,NTAPEF,IRR)                                              45
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                       48
      REWIND NTAPEF                                                        49
      CALL CAMBWB (NBODY,NWING,NTAPEE,NTAPEF,A,CL(NS),ABX,AWX)             51
      REWIND NTAPEE                                                        52
      REWIND NTAPEF                                                        53
      DO 200 J=1,NWING
  200 ALPHAW(J)=AWX(J)+ALPHAX(J)
      GO TO 360

C     CALCULATE WING CL, GIVEN WING ALPHA     KASE = 2
  210 CONTINUE

      IF (KONFIG-2) 220,230,230

C     WING ALONE CONFIGURATION
  220 CONTINUE
      CALL FSF (1,NTAPEE,IRR)                                              67
      CALL DCPD (NWING,NTAPEE,A,ALPHAW,CL)                                 69
      REWIND NTAPEE                                                        70
      GO TO 360

C     WING-BODY CONFIGURATION
  230 CONTINUE
      DO 240 J=1,NWING
  240 AWX(J)=ALPHAW(J)-ALPHAX(J)
      DO 250 J=1,NBODY
      READ (NTAPEF) (A(I),I=1,NWING)                                       84
      DO 250 I=1,NWING
  250 AWX(I)=AWX(I)-A(I)*ABX(J)
      CALL FSF (1,NTAPEE,IRR)                                              99
      CALL DCPD (NWING,NTAPEE,A,AWX,CL(NS))                                102
      REWIND NTAPEE                                                        103
      CALL FSF (1,NTAPEF,IRR)                                              104
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                       107
      REWIND NTAPEF                                                        108
      GO TO 360

C     OPTIMIZE CAMBER FOR GIVEN CL, OR FOR GIVEN CL AND CM     KASE = 3
  260 CONTINUE

      IF (CONSNT) 280,270,280
  270 CALL FSF(2,NTAPEA,IRR)                                               113
      GO TO 290
  280 CALL FSF(3,NTAPEA,IRR)                                               116
  290 CONTINUE
```

```
      IF (KONFIG-2) 300,310,310

C     WING ALONE CONFIGURATION
  300 CONTINUE
      CALL OPTMW (NWING,NTAPEA,A,B,CONSNT,CLBAR,XCPBAR,RFAREA,AREA,CL)        120
      REWIND NTAPEA                                                            121
      CALL CAMBW (NWING,NTAPEE,A,CL,ALPHAW)                                    122
      REWIND NTAPEE                                                            123
      GO TO 360

C     WING-BODY CONFIGURATION
  310 CONTINUE
      CALL OPTMWB (NWING,NBODY,NTAPEA,NTAPEF,THICK,A,B,ALPHA,ABX,ALPHAX,
     1 AREA(NS),CONSNT,CLBAR,XCPBAR,RFAREA,CL(NS))                            128
      REWIND NTAPEA                                                            129
      CALL FSF (1,NTAPEF,IRR)                                                  130
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)                           133
      REWIND NTAPEF                                                            134
      CALL CAMBWB (NBODY,NWING,NTAPEE,NTAPEF,A,CL(NS),ABX,AWX)                 136
      REWIND NTAPEE                                                            137
      REWIND NTAPEF                                                            138
      DO 350 J=1,NWING
  350 ALPHAW(J)=AWX(J)+ALPHAX(J)

C     WING ALONE OR WING-BODY CONFIGURATION
  360 CONTINUE

      IF (KONFIG-2) 370,380,380

C     WING ALONE CONFIGURATION
  370 CONTINUE
      CALL FSF (1,NTAPEB,IRR)                                                  151
      CALL CVEL (NWING,NWING,NTAPEB,A,B,C,CL,UWW,VWW,WWW)                      153
      REWIND NTAPEB                                                            154
      GO TO 390

C     WING-BODY CONFIGURATION
  380 CONTINUE
      CALL FSF (1,NTAPED,IRR)                                                  157
      CALL CVEL (NWING,NBODY,NTAPED,A,B,C,CL,UWB,VWB,WWB)                      159
      CALL FSF (2,NTAPED,IRR)                                                  161
      CALL CVEL (NWING,NWING,NTAPED,A,B,C,CL(NS),UWW,VWW,WWW)                  164
      REWIND NTAPED
                                                                               165
      CALL RITE (NFMT(6),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWB,VWB,WWB)           166
      CALL RITE (NFMT(7),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWBT,VWBT,WWBT)

C     WING ALONE OR WING-BODY CONFIGURATION                                   168
  390 CONTINUE
      CALL FSF(10,NTAPEC,IRR)                                                  170
      WRITE (NTAPEC) NPANEL                                                    171
      WRITE (NTAPEC) (CL(J),J=1,NPANEL)                                        172
      END FILE NTAPEC                                                          179
```

```
      REWIND NTAPEC                                                    180
      CALL RITE (NFMT(4),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWW,VWW,WWW)
C     COMPUTE COEFFICIENT OF PRESSURE ON WING WITH OR WITHOUT WING
C     THICKNESS                                                        181
      IF (THICK) 400,440,400
C     WING ALONE OR WING-BODY CONFIGURATION
C     WITH WING THICKNESS
  400 CONTINUE                                                         185
      CALL RITE (NFMT(5),NTAPED,NWING,NROWW,NCOLW,DUMMY,UWWT,VWWT,WWWT)
      DO 410 J=1,NWING
      U(J)=UWB(J)+UWW(J)+UWBT(J)+UWWT(J)+UWN(J)
      V(J)=VWB(J)+VWW(J)+VWBT(J)+VWWT(J)+VWN(J)
  410 W(J)=WWB(J)+WWW(J)+WWBT(J)+WWWT(J)+WWN(J)                        211
      CALL CP(O,NWING,XMACH,CPCALC,U,V,W,CPU)
      DO 420 J=1,NWING
      U(J)=UWB(J)-UWW(J)+UWBT(J)+UWWT(J)+UWN(J)
  420 W(J)=WWB(J)+WWW(J)+WWBT(J)-WWWT(J)+WWN(J)                        231
      CALL CP(O,NWING,XMACH,CPCALC,U,V,W,CPL)
      DO 430 J=1,NWING
      ALPHAU(J)=ALPHAW(J)+ALPHAT(J)
  430 ALPHAL(J)=ALPHAW(J)-ALPHAT(J)
      GO TO 480
C     WING ALONE OR WING-BODY CONFIGURATION
C     WITHOUT WING THICKNESS
  440 CONTINUE
      DO 450 J=1,NWING
      U(J)=UWB(J)+UWW(J)+UWBT(J)+UWN(J)
      V(J)=VWB(J)+VWW(J)+VWBT(J)+VWN(J)
  450 W(J)=WWB(J)+WWW(J)+WWBT(J)+WWN(J)                                268
      CALL CP(O,NWING,XMACH,CPCALC,U,V,W,CPU)
      DO 460 J=1,NWING
  460 U(J)=UWB(J)-UWW(J)+UWBT(J)+UWN(J)                                281
      CALL CP(O,NWING,XMACH,CPCALC,U,V,W,CPL)
      DO 470 J=1,NWING
      ALPHAU(J)=ALPHAW(J)
  470 ALPHAL(J)=ALPHAW(J)
C     WING ALONE OR WING-BODY CONFIGURATION
C     CALCULATE CAMBER SHAPE
  480 CONTINUE
      CALL INTPOL(NWING,NROWW,RATIOX,.5,CHORD(NS),ALPHAW,ALPHA)         295
      CALL CAMBER(NCOLW,NROWW,CHORD(NS),ALPHA,CSHAPE,CHORDL)
C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON WING               298
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD(NS),ALPHAU,ALPHA)    301
      CALL WLDM (NWING,NROWW,XP,ZP,RFAREA,AREA(NS),XBAR(NS),ZBAR(NS)
     1,ALPHA,THETA(NS),CPU,WSCLU,WSCDU,WCLU,WCDU,WCMU)                  307
      CALL INTPOL (NWING,NROWW,RATIOX,RATIOD,CHORD(NS),ALPHAL,ALPHA)    310
      CALL WLDM (NWING,NROWW,XP,ZP,RFAREA,AREA(NS),XBAR(NS),ZBAR(NS)
     1,ALPHA,THETA(NS),CPL,WSCLL,WSCDL,WCLL,WCDL,WCML)                  316
```

```
      DO 490 J=1,NCOLW
      JN=(J-1)*NROWW+(NROWW+1)/2
      JN=NBODY+JN
      DY=AREA(JN)/CHORD(JN)
      WSCD(J)=(WSCDU(J)-WSCDL(J))/DY*SEMIS
      WSCL(J)=(WSCLU(J)-WSCLL(J))/DY*SEMIS
  490 CONTINUE

      WCL=WCLU-WCLL
      WCD=WCDU-WCDL
      WCM=(WCMU-WCML)/CBAR

      IF (KONFIG .EQ. 1) GO TO 600

C     WING-BODY CONFIGURATION
      CALL CVEL (NBODY,NBODY,NTAPED,A,B,C,CL,UBB,VBB,WBB)                    337
      CALL FSF (2,NTAPED,IRR)                                                339
      CALL CVEL (NBODY,NWING,NTAPED,A,B,C,CL(NS),UBW,VBW,WBW)                342
      REWIND NTAPED
                                                                             343
      CALL RITE (NFMT(1),NTAPED,NBODY,NROWB,NTHETA,THETAB,UBB,VBB,WBB)       344
      CALL RITE (NFMT(2),NTAPED,NBODY,NROWB,NTHETA,THETAB,UBW,VBW,WBW)

C     COMPUTE COEFFICIENT OF PRESSURE ON BODY WITH OR WITHOUT THE
C     EFFECT OF WING THICKNESS                                               346
      IF (THICK) 500,520,500
  500 CONTINUE
      CALL RITE (NFMT(3),NTAPED,NBODY,NROWB,NTHETA,THETAB,UBWT,VBWT,WBWT
     1)                                                                      350
      DO 510 J=1,NBODY
      U(J)=UBB(J)+UBW(J)+UBWT(J)+UBN(J)
      V(J)=VBB(J)+VBW(J)+VBWT(J)+VBN(J)
  510 W(J)=WBB(J)+WBW(J)+WBWT(J)+WBN(J)
      GO TO 540

  520 CONTINUE
      DO 530 J=1,NBODY
      U(J)=UBB(J)+UBW(J)+UBN(J)
      V(J)=VBB(J)+VBW(J)+VBN(J)
  530 W(J)=WBB(J)+WBW(J)+WBN(J)

  540 CONTINUE
      CALL CP(1,NBODY,XMACH,CPCALC,U,V,W,CPB)

C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON BODY PANELS              394
      CALL INTPOL (NBODY,NROWB,RATIOX,RATIOD,CHORD,ALPHAB,ALPHA)             396
      CALL BLDM (NBODY,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHA,THETA,CPB
     1,BWCL,BWCD,BWCM)

C     BODY ALONE OR WING-BODY CONFIGURATION                                  398
  550 CONTINUE
      BBCL=BBCL/RFAREA
      BBCD=BBCD/RFAREA
      BBCM=BBCM/(RFAREA*CBAR)
```

```
      IF (KONFIG .EQ. 2) GO TO 600

C     WING-BODY CONFIGURATION
      BWCM=BWCM/CBAR
      BCL=BWCL+BBCL
      BCD=BWCD+BBCD
      BCM=BWCM+BBCM
      WBCL=BCL+WCL+XNNCLT
      WBCD=BCD+WCD+XNNCDT
      WBCM=BCM+WCM+XNNCMT

C     ANY CONFIGURATION
C     WRITE OUTPUT
  600 CONTINUE

      IF (KONFIG-2) 660,620,610

C     WING-BODY CONFIGURATION
  610 CONTINUE
      WRITE (NTAPEO,6010)                                          405
      GO TO 630

C     BODY ALONE CONFIGURATION
  620 CONTINUE
      WRITE (NTAPEO,6020)

C     BODY ALONE OR WING-BODY CONFIGURATION                        407
  630 CONTINUE
      WRITE (NTAPEO,6080) BBCD,BBCL,BBCM                           408
      WRITE (NTAPEO,6030)                                          409
      WRITE (NTAPEO,6036) (THETAS(I),I=1,NTHETS)                   410
      WRITE (NTAPEO,6038)                                          417
      DO 640 I=1,NBODYS
  640 WRITE (NTAPEO,6040) XB(I),(CPBB(I,J),J=1,NTHETS)
                                                                   421
      IF (KONFIG .EQ. 2) GO TO 999

C     WING-BODY CONFIGURATION
  650 CONTINUE
      WRITE (NTAPEO,6014)                                          432
      WRITE (NTAPEO,6080) BWCD,BWCL,BWCM                           433
      WRITE (NTAPEO,6032)                                          434
      CALL OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETAB,CPB)             435
      WRITE (NTAPEO,6034)                                          436
      CALL OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETAB,ALPHAB)

C     WING ALONE OR WING-BODY CONFIGURATION                        437
  660 CONTINUE

      IF (NACEL .EQ. 0) GO TO 690

C     WING ALONE OR WING-BODY CONFIGURATION WITH NACELLE(S)
      DO 680 J=1,NACEL
```

```
      NTN=XNTN(J)
      NXN=XNXN(J)
      WRITE (NTAPEO,6042) J                                            448
      WRITE (NTAPEO,6080) XNNCD(J),XNNCL(J),XNNCM(J)                   449
      WRITE (NTAPEO,6044)                                              453
      WRITE (NTAPEO,6036) (THETAN(I,J),I=1,NTN)                        454
      WRITE (NTAPEO,6038)                                              459
      DO 670 I=1,NXN
  670 WRITE (NTAPEO,6040) XN(I,J),(CPNN(I,K,J),K=1,NTN)                462
  680 CONTINUE

  690 CONTINUE
      DO 700 J=NS,NPANEL
  700 CL(J)=-CL(J)

      IF (KONFIG-2) 710,720,720

C     WING ALONE CONFIGURATION
  710 CONTINUE
      WRITE (NTAPEO,6018)                                              482
      GO TO 730

C     WING-BODY CONFIGURATION
  720 CONTINUE
      WRITE (NTAPEO,6012)

C     WING ALONE OR WING-BODY CONFIGURATION                            484
  730 CONTINUE
      WRITE (NTAPEO,6080) WCD,WCL,WCM                                  485
      WRITE (NTAPEO,6060)                                              486
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                                487
      WRITE (NTAPEO,6000) (WSCD(I),I=1,NCOLW)                          493
      WRITE (NTAPEO,6062)                                              500
      WRITE (NTAPEO,6064) (I,I=1,NCOLW)                                501
      WRITE (NTAPEO,6000) (WSCL(I),I=1,NCOLW)                          507
      WRITE (NTAPEO,6050)                                              514
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPU)                         515
      WRITE (NTAPEO,6052)                                              516
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPL)                         517
      WRITE (NTAPEO,6054)                                              518
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CL(NS))                      520
      IF (THICK) 740,750,740
  740 WRITE (NTAPEO,6056)                                              523
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAU)                      524
      WRITE (NTAPEO,6057)                                              525
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAL)                      526
  750 WRITE (NTAPEO,6058)                                              527
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAW)                      528
      WRITE (NTAPEO,6055)                                              529
      WRITE (NTAPEO,6000)(CHORDL(I),I=1,NCOLW)
                                                                       530
      IF (KONFIG .EQ. 1) GO TO 999

C     WING-BODY CONFIGURATION
```

```
      WRITE (NTAPEO,6016)                                                 539
      WRITE (NTAPEO,6080) WBCD,WBCL,WBCM
C     ANY CONFIGURATION                                                   540
  999 CONTINUE
      RETURN
C     FORMAT STATEMENTS
 6000 FORMAT(1H ,F29.5,9F10.5)
 6010 FORMAT(1H1, 48H PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY)
 6012 FORMAT(1H1, 66H PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN P
     1RESENCE OF BODY)
 6014 FORMAT(1H1, 70H INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY
     1 PANELS DUE TO WING)
 6016 FORMAT(1H1, 44H FORCES AND MOMENTS ON WING-BODY COMBINATION)
 6018 FORMAT(1H1, 38H PRESSURES, FORCES AND MOMENTS ON WING)
 6020 FORMAT(1H1, 38H PRESSURES, FORCES AND MOMENTS ON BODY)
 6030 FORMAT(1H0/32H  BODY PRESSURE COEFFICIENTS(CP))
 6032 FORMAT(1H0/37H  BODY PANEL PRESSURE COEFFICIENT(CP))
 6034 FORMAT(1H0/25H  BODY PANEL SLOPE(DR/DX))
 6036 FORMAT(1H0,4X,11HTHETA(DEG.),F14.4,9F10.4,/(F29.4,9F10.4))
 6038 FORMAT(1H ,9X,1HX)
 6040 FORMAT(1H ,F14.4,F15.5,9F10.5,/(F29.5,9F10.5))
 6042 FORMAT(1H1, 57H PRESSURES, FORCES, AND MOMENTS ON ISOLATED NACELLE
     1 PAIR ,I2)
 6044 FORMAT(1H0/34H NACELLE PRESSURE COEFFICIENTS(CP))
 6050 FORMAT(1H0/52H  UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 6052 FORMAT(1H0/52H  LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 6054 FORMAT(1H0/36H  WING PANEL PRESSURE DIFFERENCE(CL))
 6055 FORMAT(1H0/23H  WING CHORD LENGTHS(C))
 6056 FORMAT(1H0/39H  UPPER SURFACE WING PANEL SLOPE(DZ/DX))
 6057 FORMAT(1H0/39H  LOWER SURFACE WING PANEL SLOPE(DZ/DX))
 6058 FORMAT(1H0/26H  WING CAMBER SLOPE(DZ/DX))
 6060 FORMAT(1H0/40H  SPANWISE CD DISTRIBUTION(B/2Q * DD/DY))
 6062 FORMAT(1H0/40H  SPANWISE CL DISTRIBUTION(B/2Q * DL/DY))
 6064 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10,/(18X,I8,9I10))
 6080 FORMAT(1H0/6H  CD =,F10.5,10X,4HCL =,F10.5,10X,4HCM =,F10.5)
      END
```

```
     FDATE                                                                07/31/65
ASSEMBLED TEXT.

          $TEXT   FDATE                                                                              FDAT0001

                                  *     DATE AS MON YY, 19ZZ / DATE                                          DATE0010
                                  *DATE    709/90 FORTRAN LIBRARY / DATE AS MON YY, 19ZZ                     DATE0020
                                  * H. EUGENE WRIGHT  METHODS - TRANSPORT                                    DATE0030
                                  * CONVERTED TO IBMAP AUG 1963 BY D TYLER                                   DATE0040
                                  *                                                                          DATE0050
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   DATE0060
                                  *                                                                          DATE0070
                                  *     CALL DATE (A,B)                                                      DATE0080
                                  *                                                                          DATE0090
                                  * WHERE A AND B ARE THE LOCATIONS WHERE THE DATE WILL BE STORED.           DATE0100
                                  *        THE DATE FORMAT IS                                                DATE0110
                                  *                LOCATION A     MON YY                                     DATE0120
                                  *                LOCATION B     , 19ZZ                                     DATE0130
                                  *        WHERE MON REPRESENTS A  THREE CHARACTER ABBREVIATION FOR THE      DATE0140
                                  *        MONTH, AND YY AND ZZ REPRESENT THE DAY AND YEAR, RESPECTIVELY.    DATE0150
                                  *                                                                          DATE0160
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   DATE0170
                                                                                                             DATE0180
                                           ENTRY   DATE                                                      DATE0190
                                                                                                             DATE0200
                         00101    DATEBX EQU       65             IBSYS DATE                                 DATE0210
                                                                                                             DATE0220
                                  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   DATE0230

BINARY CARD (NOT PUNCHED)
      00000  000000000000        10000         LDIR                                                          DATE0240
      00001  262421632560        10000
      00002  1 00000 0 00005     10001  DATE   SAVEN   4                                                     DATE0250
      00003  0774 00 4 00000     10000
      00004  0020 00 4 00001     10000
      00005  0634 00 4 03000     10011
      00006  0634 00 4 01003     10001
      00007  0560 00 0 00101     10000         LDQ     DATEBX                                                DATE0260
      00010  4754 00 0 00000     10000         PXD     0,0                                                   DATE0270
      00011  0601 00 0 00033     10001         STO     MONTH          MAKE ZERO                              DATE0280
      00012  4763 00 0 00006     10000         LGL     6                                                     DATE0290
      00013  0601 00 0 00033     10001         STO     MONTH                                                 DATE0300
      00014  4754 00 0 00000     10000         PXD     0,0                                                   DATE0310
      00015  4763 00 0 00006     10000         LGL     6                                                     DATE0320
      00016  0737 00 4 00000     10000         PAC     0,4                                                   DATE0330
      00017  0520 00 0 00033     10001         ZET     MONTH          TEST IF OCT-DEC                        DATE0340
      00020  1 77766 4 01001     10011         TXI     *+1,4,-10                                             DATE0350
      00021  4500 00 4 00033     10001         CAL     MONTH,4        PICK UP MONTH                          DATE0360
      00022  4763 00 0 00014     10000         LGL     12             MOVE IN DAY                            DATE0370

BINARY CARD (NOT PUNCHED)
      00023  0534 00 4 00003     10001  IR4    LXA     ..0001,4                                              DATE0380
      00024  0602 60 4 00003     10000         SLW*    3,4                                                   DATE0390
      00025  4500 00 0 00050     10001         CAL     =H00, 19                                              DATE0400
      00026  4763 00 0 00014     10000         LGL     12             MOVE IN YEAR                           DATE0410
      00027  0602 60 4 00004     10000         SLW*    4,4                                                   DATE0420
      00030  4500 00 0 00101     10000         CAL     DATEBX                                                DATE0430
```

```
      FDATE                                                      07/31/65
    ASSEMBLED TEXT.

   00031  0140 00 0 02001  10011         TOV    DATE+1     TURN OFF OVERFLOW LIGHT           DATE0440
   00032  0020 00 0 02001  10011         TRA    DATE+1                                       DATE0450
                                 *                                                           DATE0460
                                 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0470
                                 *                                                           DATE0480
   00033  0 00000 0 00000  10000  MONTH  PZE    0                                            DATE0490
   00034  000041214560     10000         BCI    4,00JAN 00FEB 00MAR 00APR                    DATE0500
   00035  000026252260     10000
   00036  000044215160     10000
   00037  000021475160     10000
   00040  000044217060     10000         BCI    4,00MAY 00JUN 00JUL 00AUG                    DATE0510
   00041  000041644560     10000
   00042  000041644360     10000
   00043  000021642760     10000
   00044  000062254760     10000         BCI    4,00SEP 00OCT 00NOV 00DEC                    DATE0520
   00045  000046236360     10000

BINARY CARD (NOT PUNCHED)
   00046  000045466560     10000
   00047  000024252360     10000
                                 *                                                           DATE0530
                                 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * DATE0540
                                                                                             DATE0550

   00050  00007360C111     10000         *LORG
                     00000 01111         END                                                 DATE0560
```

```
        FDATE                                                                   07/31/65
      CONTROL DICTIONARY

               $CDICT FDATE                                                                 FDAT0002

BINARY CARD (NOT PUNCHED)
        000051000000             PREFACE      START=0,LENGTH=41,TYPE=7094,CMPLX=5
        000004000005
        252421632560     FDATE   DECK         LOC=0,LENGTH=41
        000051000000
        242163256060     DATE    REAL         LOC=2,LENGTH=0
        000000000002
        627062434623     SYSLOC  VIRTUAL      SECT. 3
        200000000000

               $DKEND FDATE                                                                 FDAT0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
        FFSF
       ASSEMBLED TEXT.

             $TEXT  FFSF                                                              FFSF0001

                                        *FSF     709/90 FORTRAN LIBRARY / FORWARD SPACE FILE                  FFSF0020
                                        * DOT TYLER    10/63                                                  FFSF0030
                                        * METHODS MEMO 7090-25                                                FFSF0040
                                        *                                                                     FFSF0050
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0060
                                        *                                                                     FFSF0070
                                        *     CALL FSF (NFILE,LTAPE,LERROR)                                   FFSF0080
                                                                                                              FFSF0090
                                                 NFILE  - NUMBER OF END-OF-FILE MARKS TO SPACE PASS.          FFSF0100
                                                          ZERO IS EQUIVALENT TO ONE.                          FFSF0110
                                                 LTAPE  - LOGICAL TAPE NUMBER.                                FFSF0120
                                                 LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.         FFSF0130
                                                          0  INDICATES SUCCESS.                               FFSF0140
                                                          N  INDICATES NUMBER OF UNSPACED FILES WHEN          FFSF0150
                                                             END-OF-TAPE SIGNAL WAS REACHED (SEE MM-23).      FFSF0160
                                        *                                                                     FFSF0170
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0180
                                                                                                              FFSF0190

BINARY CARD (NOT PUNCHED)
      00000  000000000000       10000        LDIR                                                             FFSF0200
      00001  262562266060       10000
                                                                                                              FFSF0210
                                                                                                              FFSF0220
                                        * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FFSF0230
      00002  1 00000 0 00011    10001   FSF   SAVE     (4,2,1)I                                               FFSF0240
      00003  0774 00 2 00000    10000
      00004  0774 00 1 00000    10000
      00005  0774 00 4 00000    10000
      00006  0441 00 0 00010    10001
      00007  0020 00 4 00001    10000
      00010  0 00000 0 00000    10000
      00011  0604 00 0 00010    10001
      00012  0634 00 4 05000    10011
      00013  0634 00 4 00000    10001
      00014  0634 00 4 00005    10001
      00015  0634 00 1 00004    10001
      00016  0634 00 2 00003    10001
      00017  0500 60 4 00004    10000        CLA*     4,4                                                     FFSF0250
      00020  0601 00 0 03000    10011        STO      TEMP           LOGICAL UNIT                             FFSF0260
      00021  0500 60 4 00003    10000        CLA*     3,4                                                     FFSF0270
      00022  0734 00 1 00000    10000        PAX      0,1            CTR TO XR1                               FFSF0280

BINARY CARD (NOT PUNCHED)
      00023  000000000000       00010        CALL     .FVIO.(TEMP,TEMP+1)                                     FFSF0290
      00023  0074 00 4 04000    10011
      00024  1 00002 0 01004    10011
      00025  0 00000 0 00034    10100
      00026  0 00000 0 03000    10011
      00027  0 00000 0 03001    10011
      00030  0535 00 2 03001    10011        LAC      TEMP+1,2                                                FFSF0300
      00031  0441 00 2 00001    10000        LDI      1,2                                                     FFSF0310
```

```
        FFSF
     ASSEMBLED TEXT.                                                    07/31/65

      00032  0500 00 0 03001  10011            CLA    TEMP+1                                  FFSF0320
      00033  0621 00 0 00042  10001            STA    SETA                                    FFSF0330
      00034  0621 00 0 00040  10001            STA    SETX                                    FFSF0340
      00035  4054 00   040000 10000            LFT    040000                                  FFSF0350
      00036  0020 00 0 00041  10001            TRA    LOOP                                    FFSF0360
      00037  0074 00 4 06000  10011            TSX    .OPEN,4                                 FFSF0370
      00040  5 00000 0 00000  10000     SETX   MON    **                                      FFSF0380
      00041  0074 00 4 07000  10011     LOOP   TSX    .READ,4      READ TO EOF                FFSF0390
      00042  0 00000 0 00000  10000     SETA   PZE    **,,0                                   FFSF0400
      00043  0 00046 0 00053  10101            PZE    EOF,,ERR                                FFSF0410
      00044  4 00002 0 03002  10011            IOCP   SKIM,0,2                                FFSF0420

BINARY CARD (NOT PUNCHED)
      00045  3 77777 2 00000  10000            IORTN  **,0,-1                                 FFSF0430
      00046  4500 00 0 03003  10011     ERR    CAL    SKIM+1                                  FFSF0440
      00047  4340 00 0 00063  10001            LAS    CODE         IF TAPEND                  FFSF0450
      00050  0020 00 0 00041  10001            TRA    LOOP                                    FFSF0460
      00051  0020 00 0 00057  10001            TRA    ERRA                                    FFSF0470
      00052  0020 00 0 00041  10001            TRA    LOOP                                    FFSF0480
      00053  2 00001 1 00041  10001     EOF    TIX    LOOP,1,1                                FFSF0490
      00054  0534 00 4 00005  10001            LXA    ..0001,4                                FFSF0500
      00055  0600 60 4 00005  10000            STZ*   5,4          SET N=0                    FFSF0510
                       00056                   RETURN FSF                                     FFSF0520
      00057  0754 00 1 00000  10000     ERRA   PXA    0,1          SET N TO CTR OF            FFSF0530
      00060  0534 00 4 00005  10001            LXA    ..0001,4       UNSKIPPED FILES          FFSF0540
      00061  0601 60 4 00005  10000            STO*   5,4                                     FFSF0550
      00062  0020 00 0 00056  10001            TRA    ERRA-1                                  FFSF0560
      00063  632147254524     10000     CODE   BCI    1,TAPEND                                FFSF0570
                       00064                   LORG                                           FFSF0580
                                        ERASE  CONTRL ER                                      FFSF0590
      00064  100000000064     00001            USE    ER                                      FFSF0600
      00064  200000000002     00001     TEMP   BSS    2                                       FFSF0610
      00066  200000000002     00001     SKIM   BSS    2                                       FFSF0620
      00070  200000000020     00001            BSS    16                                      FFSF0630

BINARY CARD (NOT PUNCHED)
                       00000  01111            END                                            FFSF0640
```

```
       FFSF                                                   07/31/65
   CONTROL DICTIONARY
                                                                                FFSF0002
          $CDICT FFSF

BINARY CARD (NOT PUNCHED)
           000110000000              PREFACE   START=0,LENGTH=72,TYPE=7094,CMPLX=5
           000004000005
           262662265060      FFSF    DECK      LOC=0,LENGTH=72
           000110000000
           266226606060      FSF     REAL      LOC=2,LENGTH=0
           000000000002
           255121622560      ERASE   REAL      LOC=64,LENGTH=20
           000024000064
           332665314633      .FVIO.  VIRTUAL   SECT. 4,CALL
           200000100000
           627062434623      SYSLOC  VIRTUAL   SECT. 5
           200000000000
           334647254560      .OPEN   VIRTUAL   SECT. 6
           200000000000
           335125212460      .READ   VIRTUAL   SECT. 7
           200000000000

          $DKEND FFSF                                                           FFSF0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
        =FSF                                                     07/31/65
      SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

 CLASS  SYMBOL    VALUE   REFERENCES

        CODE      00063   47
        EOF       00053   43
        ERRA      00057   51,62
        ERR       00046   43
 LCTR   ER
        FSF       00002   56
        ..0001    00005   14,15,16,54,60
        ..0002    00007   6,11
        ..0003    00011   2
        LOOP      00041   36,50,52,53
 LCTR    BLCTR
 QUAL    UNQS
 LCTR   //
        SETA      00042   33
        SETX      00040   34
        SKIM      00066   44,46
        TEMP      00064   20,26,27,30,32

REFERENCES TO VIRTUAL SYMBOLS.

        .FVID.        4   23
        .OPEN         6   37
        .READ         7   41
        SYSLOC        5   12
```

```
             $TEXT  FFSR                                                                      FFSR0001

                                               *     FORWARD SPACE RECORD / FSR                                      FSR00020
                                               * JIM STOKER   TRANSPORT - 4/61                                       FSR00030
                                               * CONVERTED TO IBMAP BY DOT TYLER - 10/63                             FSR00040
                                               * METHODS MEMO 7090-24                                                FSR00050
                                               *                                                                     FSR00060
                                               * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00070
                                               *                                                                     FSR00080
                                               *     CALL FSR (NREC,LTAPE,LERROR)                                    FSR00090
                                               *                                                                     FSR00100
                                               *          NREC - NUMBER OF FORTRAN LOGICAL RECORDS TO SPACE FORWARD. FSR00110
                                               *         LTAPE - LOGICAL TAPE NUMBER.                                FSR00120
                                               *        LERROR - LOCATION IN WHICH THE ERROR CODE IS STORED.         FSR00130
                                               *                 O  INDICATES SUCCESS.                               FSR00140
                                               *                 N  INDICATES NUMBER OF UNSPACED RECORDS WHEN AN     FSR00150
                                               *                    END-OF-FILE MARK WAS FOUND.  TAPE LEFT SPACED IN FSR00160
                                               *                    FRONT OF EOF.                                    FSR00170
                                               *                                                                     FSR00180
                                               *     CALL      FSPR(NPREC,LTAPE,LERROR)                              FSR00190
                                               *                                                                     FSR00200
                                               *        NPREC - NUMBER PHYSICAL RECORDS TO SPACE FORWARD.            FSR00210
                                               *                                                                     FSR00220
                                               * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00230
                                                                                                                     FSR00240
                                                      ENTRY    FSR                                                   FSR00250
                                                      ENTRY    FSPR                                               M3 FSR00260
                                                                                                                     FSR00270
                                                                                                                     FSR00280
                                               * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00290

BINARY CARD (NOT PUNCHED)
       00000  000000000000     10000          LDIR                                                                   FSR00300
       00001  262662516060     10000
       00002  0600 00 0 04004  10011   FSPR   STZ      CODE                                                          FSR00310
       00003  0020 00 0 01002  10011          TRA      *+2                                                           FSR00320
       00004  4625 00 0 04004  10011   FSR    STL      CODE                                                          FSR00330
       00005  1 00000 0 00014  10001   TURN   SAVEN    4,2,1,I                                                       FSR00340
       00006  0774 00 2 00000  10000
       00007  0774 00 1 00000  10000
       00010  0774 00 4 00000  10000
       00011  0441 00 0 00013  10001
       00012  0020 00 4 00001  10000
       00013  0 00000 0 00000  10000
       00014  0604 00 0 00013  10001
       00015  0634 00 4 06000  10011
       00016  0634 00 4 00010  10001
       00017  0634 00 1 00007  10001
       00020  0634 00 2 00006  10001
       00021  0500 60 4 00004  10000          CLA*     4,4                                                           FSR00350
       00022  0601 00 0 04000  10011          STO      TEMP                                                          FSR00360

BINARY CARD (NOT PUNCHED)
       00023  0500 60 4 00003  10000          CLA*     3,4                                                           FSR00370
```

```
        FFSR                                                             07/31/65
     ASSEMBLED TEXT.

    00024  0734 00 1 00000   10000          PAX     0,1                                        FSR00380
    00025  0100 00 0 00063   10001          TZE     WIND                                       FSR00390
    00026  000000000000      00010          CALL    .FVIO.(TEMP,TEMP+1)                        FSR00400
    00026  0074 00 4 05000   10011
    00027  1 00002 0 01004   10011
    00030  0 00000 0 00047   10100
    00031  0 00000 0 04000   10011
    00032  0 00000 0 04001   10011
    00033  0535 00 2 04001   10011          LAC     TEMP+1,2                                   FSR00410
    00034  0636 00 2 00044   10001          SCA     SETA,2                                     FSR00420
    00035  0636 00 2 00042   10001          SCA     SETB,2                                     FSR00430
    00036  0441 00 2 00001   10000          LDI     1,2                                        FSR00440
    00037  4054 00  040000   10000          LFT     040000                                     FSR00450
    00040  0020 00 0 00043   10001          TRA     READL                                      FSR00460
    00041  0074 00 4 07000   10011          TSX     .OPEN,4                                    FSR00470
    00042  5 00000 0 00000   10000    SETB  MON     **                                         FSR00480
    00043  0074 00 4 10000   10011    READL TSX     .READ,4                                    FSR00490
    00044  0 00000 0 00000   10000    SETA  PZE     **,,0                                      FSR00500

BINARY CARD (NOT PUNCHED)
    00045  0 00050 0 00060   10101          PZE     EOF,,ERR                                   FSR00510
    00046  4 00002 0 04002   10011          IOCP    SKIM,0,2                                   FSR00520
    00047  3 77777 2 00000   10000          IORTN   **,0,-1                                    FSR00530
    00050  0534 00 4 04002   10011    ERR   LXA     SKIM,4                                     FSR00540
    00051  0520 00 0 04004   10011          ZET     CODE                                       FSR00550
    00052  7 00000 4 00043   10001          TXL     READL,4,0                                  FSR00560
    00053  2 00001 1 00043   10001          TIX     READL,1,1                                  FSR00570
    00054  4754 00 0 00000   10000          PXD     0,0                                        FSR00580
    00055  0534 00 4 00010   10001    DONE  LXA     ..0001,4                                   FSR00590
    00056  0601 60 4 00005   10000          STO*    5,4                                        FSR00600
                     00057                  RETURN  TURN                                       FSR00610
    00060  0074 00 4 11000   10011    EOF   TSX     .NDSEL,4                                   FSR00620
    00061  0 00005 2 00000   10000          PZE     0,2,5                                      FSR00630
    00062  0761 00 0 00000   10000          NOP                                                FSR00640
    00063  0754 00 1 00000   10000    WIND  PXA     0,1                                        FSR00650
    00064  0020 00 0 00055   10001          TRA     DONE                                       FSR00660
                                    *                                                          FSR00670
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00680
                                    *                                                          FSR00690
                                      ERASE CONTRL  ER                                         FSR00700
    00065  100000000065      00001          USE     ER                                         FSR00710
    00065  200000000002      00001    TEMP  BSS     2                                          FSR00720
    00067  200000000002      00001    SKIM  BSS     2                                          FSR00730

BINARY CARD (NOT PUNCHED)
    00071  200000000001      00001    CODE  BSS     1                                          FSR00740
    00072  200000000017      00001          BSS     15                                         FSR00750
                                    *                                                          FSR00760
                                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FSR00770
                                                                                               FSR00780
                     00000   01111          END                                                FSR00790
```

```
          FFSR                                                                        07/31/65
      CONTROL DICTIONARY

             $CDICT FFSR                                                                          FFSR0002

BINARY CARD (NOT PUNCHED)
         000111000000                PREFACE       START=0,LENGTH=73,TYPE=7094,CMPLX=5
         000004000005
         262662516060       FFSR     DECK          LOC=0,LENGTH=73
         000111000000
         266247516060       FSPR     REAL          LOC=2,LENGTH=0
         000000000002
         266251606060       FSR      REAL          LOC=4,LENGTH=0
         000000000004
         255121622560       ERASE    REAL          LOC=65,LENGTH=20
         000024000065
         332665314633       .FVIO.   VIRTUAL       SECT. 5,CALL
         200000100000
         627062434623       SYSLOC   VIRTUAL       SECT. 6
         200000000000
         334647254560       .OPEN    VIRTUAL       SECT. 7
         200000000000
         335125212460       .READ    VIRTUAL       SECT. 8
         200000000000
         334524622543       .NDSEL   VIRTUAL       SECT. 9
         200000000000

             $DKEND FFSR                                                                          FFSR0003

NO MESSAGES FOR THIS ASSEMBLY
```

```
      SUBROUTINE FLOOUT(IC)
      COMMON /FLOV1/KACE,NPANEL,NBCDY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM
      DIMENSION NROW(2)
      COMMON /FLOV4/COSARA,SINARA,XMACH2,BT2
      COMMON /FLOV5/N1,N2,N3,N4
      COMMON /THICK/THKW,ARA,CPCALC,CAMN,POP
      COMMON /INTGA/GP(3,500),VLS(4,500)
      COMMON DUM(9),N6
 1001 FORMAT(1H1,86HVELOCITY COMPONENTS AND PRESSURE COEFFICIENTS DUE TC
     $ WING PANEL PRESSURE SINGULARITIES)
 1002 FORMAT(1H1,65HVELCCITY COMPCNENTS AND PRESSURE CCEFFICIENTS DUE TC
     $ WING SOURCES)
 1003 FORMAT(1H1,86HVELOCITY COMPCNENTS AND PRESSURE CCEFFICIENTS DUE TC
     $ BODY PANEL PRESSURE SINGULARITIES)
 1004 FORMAT(1H1,83HVELOCITY COMPONENTS AND PRESSURE CCEFFICIENTS DUE TC
     $ BODY LINE SOURCES AND DOUBLETS)
 1005 FORMAT(1H0,//,66X,23HPERTURBATICN CCMPCNENTS)
 1006 FORMAT(1H0,7X,1HX,14X,1HY,14X,1HZ,16X,1HU,14X,1HV,14X,1HW,16X,2HCP
     $/(1H0,3F15.6,5X,3F15.7,5X,F15.7))
      IF(POP.EQ.0.) RETURN                                          FLCOUT
      IF(IC.GT.500) IC=500
      REWINC N1                                                              5
      REWIND N2                                                              6
      REWIND N3                                                              7
      REWIND N4                                                              8
      IF(KACE.EQ.2) GO TO 32
      IF(THKW.EQ.0.) GC TC 15
      DO 10 J=1,IC
      READ(N1)  GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)          17
   10 CONTINUE
      WRITE(N6,1002)                                                        26
      NP=1
      GO TC 100
   15 IF(CAMN.NE.0.) GO TC 31
      DO 20 J=1,IC
      READ(N2)  GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)          33
   20 CONTINUE
      WRITE(N6,1001)                                                        42
      NP=2
      GO TO 100
   21 CONTINUE
      DO 30 J=1,IC
      READ(N3)  GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)          47
   30 CONTINUE
      WRITE(N6,1003)                                                        56
      NP=3
      GC TC 100
   31 IF(KACE.NE.3) RETURN
   32 DO 40 J=1,IC
      READ(N4)  GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)          64
   40 CONTINUE
      WRITE(N6,1004)                                                        73
      NP=4
```

```
100 CONTINUE
    WRITE(N6,1005)                                                      74
    DO 110 J=1,IC
    U=VLS(1,J)
    V=VLS(2,J)
    W=VLS(3,J)
    IF (CPCALC-1.) 320,330,335
320 CONTINUE
C       LINEAR CP FORMULA
    CP = -2.*U
    GO TO 340
330 CONTINUE
C       NON-LINEAR CP FORMULA
    CP = -2.*U+BT2*U*U-V*V-W*W
    GO TO 340
C        **EXACT**  ISENTROPIC CP FORMULA
335 Q2=(U+COSARA)**2+V*V+(W+SINARA)**2
    CP = 1.42857/XMACH2*((1.+.2*XMACH2*(1.-Q2))**3.5-1.)               87
340 CONTINUE
110 VLS(4,J)=CP
    WRITE(N6,1006)((GP(I,J),I=1,3),(VLS(I,J),I=1,4),J=1,IC)             92
    REWIND N1                                                           106
    REWIND N2                                                           107
    REWIND N3                                                           108
    REWIND N4                                                           109
    GO TO(15,21,31,120),NP
120 RETURN
    END
```

```
      SUBROUTINE FLOVIZ
C********MAIN PROGRAM FOR STREAMLINES
      COMMON /GRIDA/MODE,DX,DY,DZ,IX,IY,IZ,XO,YO,ZO,XA,YA,ZA,LIT
      COMMON /INTGA/GP(3,500),VLS(4,500)
      COMMON /FLOV1/KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM
      COMMON /FLOV2/X,Y,Z,NPART,ALPHAS,THETA,XBB,R,WT,T,TC,SST,CHORD
      COMMON /FLOV4/COSARA,SINARA,XMACH2,BT2
      COMMON /FLOV5/N1,N2,N3,N4
      COMMON /THICK/THKW,ARA,CPCALC,CAMN,POP
      COMMON /TODO/ DXMAX,DXMIN,PRINT
      COMMON DUM(4),N8,DM(3),N5,N6
      INTEGER DUM,DM
      DIMENSION X(210,3,4),Y(210,3,4),Z(210,3,4),NROW(2),NPART(210)
      DIMENSIONALPHAS(210),THETA(210)
      DIMENSION XBB(50),R(50),WT(120),
      DIMENSION T(50),TC(50),SST(210) ,CHORD(210)
      DATA GRID/6HGRIDS /,STREAM/6HSTREAM/,POINT/6HPOINTS/
 1001 FORMAT(A6)
 1002 FORMAT(1H0, 39HFLOW VIZUALIZATION DONE LAST CARD READ=A6)
 1003 FORMAT(1H1,A6)
 1004 FORMAT(6F10.5,A1)
 1005 FORMAT(1H0, 9HCOMPUTE**,I3,7H**GRIDS)
 1006 FORMAT(7F10.5)
 1007 FORMAT(1H0,11HSTREAMLINES)
 1008 FORMAT(8F10.5)
 1009 FORMAT(1H0,6HPOINTS)
 1010 FORMAT(1H0,6X,1HX,12X,1HY,12X,1HZ,11X,2HUP,11X,2HVP,11X,2HWP,
     $10X,1HU,12X,1HV,12X,1HW,11X,2HCP)
 1011 FORMAT(1X,10F13.6)
 1013 FORMAT(1H0,/1X,13HINITIAL POINT,2X,2HX=F10.5,2X,2HY=F10.5,2X,2HZ=F
     $10.5)
 1014 FORMAT(1H0,17HINITIAL STEPSIZE=F10.5)
 1015 FORMAT(1H1)
 1016 FORMAT(1H0,49HFOR THIS FLOW VISUALIZATION PRESSURE COEFFICIENTS/
     $         1X,24HWILL BE CALCULATED USING,8(1H-),19HA LINEAR CP FORMU
     $LA)
 1017 FORMAT(1H0,49HFOR THIS FLOW VISUALIZATION PRESSURE COEFFICIENTS/
     $         1X,24HWILL BE CALCULATED USING,8(1H-),23HA NON-LINEAR CP F
     $ORMULA)
 1018 FORMAT(1H0,49HFOR THIS FLOW VISUALIZATION PRESSURE COEFFICIENTS/
     $         1X,24HWILL BE CALCULATED USING,8(1H-),32HAN -EXACT- ISENTR
     $OPIC CP FORMULA)
 1019 FORMAT(1H-  ,1X,81HEFFECTS DUE TO CAMBER AND ANGLE OF ATTACK DUE T
     $O WING PANELS WILL NOT BE INCLUDED)
 1020 FORMAT(1H0,90HVELOCITY COMPONENTS DUE TO WING PANEL PRESSURE SINGU
     $LARITIES *ARE* CALCULATED IN THE FIELD)
 1021 FORMAT(1H0,85HVELOCITY COMPONENTS DUE PANEL PRESSURE SINGULARITIES
     $ *ARE NOT CALCULATED IN THE FIELD)
 1022 FORMAT(1H0,80HVELOCITY COMPONENTS DUE TO BOTH WING AND BODY PANEL
     $PRESSURE SINGULARITIES *ARE*/1X,23HCALCULATED IN THE FIELD)
      N1=DUM(3)
      N2=DUM(4)
      N3=DM(1)
```

```
08/08/67
     FLOVIZ      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      N4=DM(2)
      REWIND N1                                                   5
      REWIND N2                                                   6
      REWIND N3                                                   7
      REWIND N4                                                   8
      CALL ALLIN(N8)                                              9
      COSARA=COS(ARA)                                            11
      SINARA=SIN(ARA)                                            12
      XMACH2=XMACH*XMACH
      BT2=XMACH2-1.
      READ(N5,1006) CPCALC,CAMN,POP                              13
      IF(POP.EQ.0.) GO TO 50
      IF(CAMN.EQ.0.) WRITE(N6,1020)                              17
      IF(CAMN.EQ.1.) WRITE(N6,1021)                              19
      IF(CAMN.EQ.2.) WRITE(N6,1022)                              21
   50 CONTINUE
      IF(CPCALC-1.) 5,10,15
    5 WRITE(N6,1016)                                             25
      GO TO 20
   10 WRITE(N6,1017)                                             27
      GO TO 20
   15 WRITE(N6,1018)                                             29
   20 CONTINUE
      IF(CAMN.EQ.1.) WRITE(N6,1019)                              31
    1 CONTINUE
      READ(N5,1001) CUEWRD                                       33
      IF(CUEWRD.EQ.GRID) GO TO 200
      IF(CUEWRD.EQ.STREAM) GO TO 300
      IF(CUEWRD.EQ.POINT) GO TO 400
  100 WRITE(N6,1002) CUEWRD                                      43
      RETURN
  200 WRITE(N6,1003) CUEWRD                                      44
      READ(N5,1004) XNG                                          45
      NGR=XNG
      WRITE(N6,1005) NGR                                         46
      DO 215 JJ=1,NGR
      READ(N5,1004) XO,YO,ZO,XA,YA,ZA,LIT                        49
      READ(N5,1006) DX,DY,DZ,XN,YN,ZN,AMD                        51
      IX=XN
      IY=YN
      IZ=ZN
      MODE=AMD
      CALL GRIDS                                                 56
  215 CONTINUE
      GO TO 1
  300 CONTINUE
      WRITE(N6,1015)                                             61
      WRITE(N6,1007)                                             62
      READ(N5,1006) DXMAX,DXMIN,PRINT                            63
      READ(N5,1004) XNS                                          64
      NS=XNS
      DO 315 JJ=1,NS
      READ(N5,1006) XS,YS,ZS,XMAX,XMIN,XDELT                     68
      WRITE(N6,1013)  XS,YS,ZS                                   69
```

```
      WRITE(N6,1014) XDELT                                              70
      CALL STRMI(XS,YS,ZS,XMAX,XMIN,XDELT,1)                            71
      CALL STRMI(XS,YS,ZS,XMAX,XMIN,XDELT,2)                            73
  315 CONTINUE
      GO TO 1
  400 CONTINUE
C     POINTS COMPUTATION
      REWIND N1                                                         78
      REWIND N2                                                         79
      REWIND N3                                                         80
      REWIND N4                                                         81
      WRITE(N6,1015)                                                    82
      WRITE(N6,1009)                                                    83
      WRITE(N6,1010)                                                    84
      READ(N5,1004) XP                                                  85
      ISTR=0
      NP=XP
      DO 415 JJ=1,NP
      READ(N5,1004) XF,YF,ZF                                            90
      CALL LACKEY(ISTR,          XF,YF,ZF,U,V,W,CPL)                    91
      COSA=U-COSARA
      COSB=V
      COSC=W-SINARA
      WRITE(N6,1011) XF,YF,ZF,COSA,COSB,COSC,U,V,W,CPL                  93
  415 CONTINUE
       CALL FLOOUT(NP)                                                  97
      GO TO 1
      END
```

```
      SUBROUTINE FORCES
C     CONTROL ROUTINE FOR FORCES LINK
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     A               RWT(20),ALPHA(210),ALPHAB(210),ALPHAS(210),ALPHAT(1
     A               10),ALPHAN(110),ALPHAX(110),AWS(110)
     A               ,ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW,ARAS,AT,AAA
      COMMON /BVAR/  B(210)
     B               ,BBCL,BBCD,BBCM
      COMMON /CVAR/  C(210),CHORD(210),CL(210),CPBB(55,10),CPNN(55,10,2)
     C               ,CLS(210)
     C               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDM
      COMMON /DVAR/  D(210),DZDXB(55)
     D               ,DADEG,DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     N               ,NS,NPANEL,NACEL,NROWB,NROWW,NCCLW,NBODYS,NTHETA,NT
     N               HETS,NXLE,NRG,NPOLAR,NCLX
      COMMON /PVAR/  POLAR
      COMMON /RVAR/  R(55),RN(55,2)
     R               ,RATIOX,RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON /TVAR/  TITLE(12),THETA(210),THETAB(10),THETAS(9),THETAN(10
     T               ,2),TCL(10),TCD(10),TCM(10)
     T               ,THICK,TWIST
      COMMON /UVAR/  UBWT(100),UWBT(110),UWWT(110),UNCL(210,2)
      COMMON /VVAR/  VBWT(100),VWBT(110),VWWT(110),VNCL(210,2)
     V               ,VOUT
      COMMON /WVAR/  WBWT(100),WWBT(110),WWWT(110),WNCL(210,2)
      COMMON /XVAR/  XBAR(210),XC(210),XNI(2),XNXN(2),XNTN(2),XB(55),XN(
     X               55,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(10),XCD
     X               (10)
     X               ,XNACEL,XP,XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON /ZVAR/  ZBAR(210),ZC(210),ZNI(2),ZDELTA(55),ZDN(55,2)
     Z               ,ZP,ZA
C     DEFINE CONSTANTS AND READ INPUT DATA
C     KONFIG = 1     WING ALONE CONFIGURATION
C     KONFIG = 2     BODY ALONE CONFIGURATION
C     KONFIG = 3     WING-BODY CONFIGURATION
C     CASE = 1.      CALCULATE ALPHA, GIVEN CL
C     CASE = 2.      CALCULATE CL, GIVEN ALPHA
C     CASE = 3.      OPTIMIZE WING
```

```
C     CPCALC =0.      LINEAR CP
C     CPCALC =1.      NON-LINEAR CP
C     CPCALC =2.      EXACT CP

C     POLAR = 0.      POLARS NOT REQUESTED,OMIT DELTA ALPHA CARDS
C     POLAR = 1.      POLARS REQUESTED,INSERT DELTA ALPHA CARDS

C     THICK =0.       WING THICKNESS PRESSURES NOT TO BE ADDED
C     THICK =1.       WING THICKNESS PRESSURES TO BE ADDED
C     THICK =2.       WING THICKNESS PRESSURES TO BE ADDED USING
C                     RIEGELS CORRECTION RULE

C     VOUT = 0.       VELOCITY COMPONENTS NOT TO BE PRINTED
C     VOUT = 1.       VELOCITY COMPONENTS TO BE PRINTED

C     XNACEL =0.      EFFECTS OF NACELS NOT INCLUDED
C     XNACEL =X.      EFFECTS OF X PAIRS OF NACELLES, GECMETRY EXPECTED
C     XNACEL =-X.     EFFECTS OF X PAIRS OF NACELLES, GECMETRY SAME AS
C                     PREVIOUS CASE

C     RFAREA =0.      WING REFERENCE AREA TO BE CALCULATED
C     RFAREA =X.      WING REFERENCE AREA TO BE SPECIFIED

C     XP,ZP           X AND Z COORDINATES OF POINT ABOUT WHICH MOMENTS
C                     ARE TO BE COMPUTED

C     CBAR            REFERENCE CHORD LENGTH USED IN PITCHING
C                     MOMENT CALCULATIONS
C                     CBAR =1. IF NOT SPECIFIED

C     SEMIS           WING SEMI-SPAN USEDIN SPANWISE CD AND CL
C                     CALCULATIONS
C                     SEMIS =1. IF NOT SPECIFIED

      KONFIG=KACE
      NPANEL=NWING+NBODY
      NS=NBODY+1
      CDR=57.2957795
      IF (KONFIG-2) 5,10,5

C     READ CONFIGURATION GEOMETRY
 5    CONTINUE
      READ  (NTAPEC) NBODY,NWING,XMACH,SYM,KACE                                6
      READ  (NTAPEC) (I,XBAR(I),YBAR(I),ZBAR(I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),ALPHAS(I),CHORD(I),I=1,NPANEL),NRG,(NRCW(I),I=1,NRG),RA
     2TIOX                                                                     10
      REWIND NTAPEC                                                            34

 10   CONTINUE

C     READ CASE DATA
      READ (NTAPEI,5000) TITLE                                                 35
```

```
      READ (NTAPEI,5010) CASE,CPCALC,POLAR,THICK,VOUT                   37
      IPOLAR=0
      NACEL=0
      IF (CASE) 20,999,20
 20   CONTINUE
      READ (NTAPEI,5010) RFAREA,XP,ZP,CBAR,SEMIS
                                                                        41
      NTHETS=9
      THETAS(1)=0.
      DO 50 I=2,NTHETS
      THETAS(I)=THETAS(I-1)+22.5
   50 CONTINUE

      IF (CBAR .EQ. 0.) CBAR=1.
      IF (SEMIS .EQ. 0.)SEMIS=1.

      IF (VOUT) 75,85,75
 75   CONTINUE
      DO 80 J=1,9
 80   NFMT(J)=J
      GO TO 100
   85 CONTINUE
      DC 90 J=1,9
   90 NFMT(J)=0
  100 CONTINUE
      KASE=CASE
      IF (RFAREA) 120,120,140
  120 IF (KONFIG-2) 130,125,130
  125 RFAREA=1.0
      GO TO 140
  130 RFAREA=0.0
      DO 135 J=NS,NPANEL
  135 RFAREA=RFAREA+AREA(J)
  140 CONTINUE

C     WRITE CASE DATA
      WRITE (NTAPEO,6002) TITLE,DATE                                    88
      CALL INOUT(NTAPEO,KASE,CPCALC,POLAR,THICK,VOUT,XMACH,SEMIS,
     1XP,ZP,CBAR,RFAREA,SYM)
                                                                        91
  200 CONTINUE

C     ANY CONFIGURATION
      IF (KONFIG .EQ. 1) GO TO 240

C     BODY ALONE OR WING-BODY CONFIGURATION
      NROWB=NROW(1)

      IF (KONFIG-2) 210,210,220

C     BODY ALONE CONFIGURATION
  210 READ (NTAPEC) NBODY,NWING,XMACH,SYM,KACE                          98
      GO TO 230
```

```
C       WING-BODY CONFIGURATION
  220 CALL FSF(3,NTAPEC,IRR)                                              104
  230 CONTINUE
      READ (NTAPEC) ALPHAA                                                105
      READ (NTAPEC) NBODYS,NXLE,ZA,(XB(I),I=1,NBODYS)                     106
      READ (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)

C       INPUT BODY VARIABLES                                              115
      READ (NTAPEI,5010) ARB                                              123
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NBODYS,R)                           124
      READ (NTAPEC) (ZDELTA(I),I=1,NBODYS)                                125
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NBODYS,ZDELTA)                      132
      REWIND NTAPEC
                                                                          133
      ARA=-ALPHAA+ARB/CDR
      WRITE (NTAPEO,6094) ZA
                                                                          134
      IF (KONFIG .EQ. 2) GO TO 250

C       WING-BODY CONFIGURATION
      CALL BCAM(NBODYS,NROWB,XB,ZDELTA,XC,DZDXB,ACB)                      138
      CALL BTHICK(NBODYS,NBODY,NROWB,XB,R,XC,ALPHA,ALPHAS)

C       WING ALONE OR WING-BODY CONFIGURATION                             140
  240 CONTINUE
      NROWW=NROW(NRG)
      NCOLW=NWING/NROWW

      IF (KONFIG .EQ. 3) GO TO 250

C       WING ALONE CONFIGURATION
      ALPHAA=0.
      ARA=0.

C       ANY CONFIGURATION
  250 CONTINUE
      CALL FSF(7,NTAPEC,IRR)                                              149
      WRITE (NTAPEC) THICK,ARA                                            150
      END FILE NTAPEC                                                     151
      REWIND NTAPEC                                                       152
      ALPHAD=ALPHAA*CDR
      ARADEG=ARA*CDR
      WRITE (NTAPEO,6090) ALPHAD                                          153
      WRITE (NTAPEO,6092) ARADEG
                                                                          154
      IF (KONFIG.EQ.2) GO TO 420

C       WING ALONE OR WING-BODY CONFIGURATION
C       INPUT WING VARIABLES
      GO TO (401,404,409),KASE
  401 CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,CL(NS))                       161
      DO 402 J=NS,NPANEL
  402 CL(J)=-CL(J)
```

```
      GO TO 411
  404 READ (NTAPEI,5010) ARW,TWIST                                           172
      IF (TWIST .NE. 0.) GO TO 406
      DO 405 J=1,NCOLW
  405 ARWT(J)=0.0
      GO TO 407
  406 READ (NTAPEI,5010) (ARWT(I),I=1,NCOLW)                                 184
  407 CONTINUE
      CALL FSF (2,NTAPEC,IRR)                                                192
      CALL READ (NTAPEI,NTAPEO,NTAPEC,NWING,ALPHA)                           194
      REWIND NTAPEC                                                          195
      DO 408 J=1,NWING
      JJ=(J-1)/NROWW+1
  408 ALPHAW(J)=ALPHA(J)-ARA-(ARW+ARWT(JJ))/CDR
      GO TO 411
  409 READ (NTAPEI,5010) CONSNT,CLBAR,XCPBAR                                 206
      IF (CONSNT .NE. 0.) GO TO 410
      WRITE (NTAPEO,6082) CLBAR                                              209
      GO TO 411
  410 WRITE (NTAPEO,6082) CLBAR                                              211
      WRITE (NTAPEO,6084) XCPBAR                                             212
  411 CONTINUE

C     WING ALONE OR WING-BODY CONFIGURATION
      IF (THICK) 440,420,440

  420 CONTINUE
      CALL FSF(8,NTAPEC,IRR)                                                 216
      END FILE NTAPEC                                                        217
      REWIND NTAPEC                                                          218
      IF (KONFIG-2) 470,425,472

C     WING-BODY CONFIGURATION WITHOUT WING THICKNESS
  425 CONTINUE
      DO 430 J=1,NBODY
  430 ALPHA(J)=0.
      GO TO 450

C     WING ALONE OR WING-BODY CONFIGURATION
  440 CONTINUE
      CALL TVEL(A,B,C,D,ALPHAT,UBWT,VBWT,WBWT,UWWT,VWWT,WWWT,
     $ALPHA,CHORD,THICK,NROW)                                                230
      REWIND NTAPEA                                                          231
      REWIND NTAPEB
                                                                             232
      IF (KONFIG .EQ. 1) GO TO 470

C     WING-BODY CONFIGURATION
  450 DO 460 J=1,NBODY
      JJ=J-((J-1)/NROWB)*NROWB
      ALPHAB(J)=ALPHAS(J)-(ARA    -ACB(JJ))*COS(THETA(J))                    243
      ABX(J)=-ALPHA(J)
  460 CONTINUE
```

```
C     ANY CONFIGURATION
  470 CONTINUE
      DO 471 J=1,NWING
      UWBT(J)=0.
      VWBT(J)=0.
      WWBT(J)=0.
  471 ALPHAX(J)=0.

      IF (KONFIG.NE.1) GO TO 472
      CALL FSF(9,NTAPEC,IRR)                                          265
      END FILE NTAPEC                                                 266
      REWIND NTAPEC                                                   267
      GO TO 480

  472 CONTINUE
C     BODY ALONE OR WING-BODY CONFIGURATION
      CALL KARMOR(NBODYS,NTHETS,  0.,0.,0.,CPCALC,VOUT,ARA
     1,XP,ZP,RFAREA,XB,R,THETAS,ZDELTA,XC,YC,ZC,THETA,UWBT,VWBT,WWBT
     2,ALPHAX,CPBB,BBCL,BBCD,BBCM)
                                                                      270
  480 CONTINUE

C     ANY CONFIGURATION
  500 CONTINUE

      CALL FCALC

C     ANY CONFIGURATION
C     READ ADDITIONAL POLARS                                          273
      IF (POLAR) 510,10,510

  510 CONTINUE
      READ (NTAPEI,5010) DADEG
                                                                      276
      IF (DADEG) 520,10,520

  520 CONTINUE
      IPOLAR=1
      DARAD=DADEG/CDR
      ARA=ARA+DARAD
      ARADEG=ARA*CDR
      WRITE (NTAPEO,6002) TITLE                                       280
      WRITE (NTAPEO,6008) DADEG                                       282
      WRITE (NTAPEO,6092) ARADEG
                                                                      283
      IF (KONFIG .EQ. 2) GO TO 550

C     WING ALONE OR WING-BODY CONFIGURATION
      DO 530 I=1,NWING
  530 ALPHAW(I)=ALPHAW(I)-DARAD

      IF (KONFIG .EQ. 1) GO TO 550

C     WING-BODY CONFIGURATION
```

```
      DO 540 I=1,NBODY
  540 ALPHAB(I)=ALPHAB(I)-DARAD*COS(THETA(I))

C      ANY CONFIGURATION
  550 CONTINUE

      IF (KONFIG-2) 470,472,470

 999  CONTINUE
      RETURN


 5000 FORMAT(12A6)
 5010 FORMAT(7F10.0)
 6002 FORMAT(1H1,12A6,30X,2A6)
 6008 FORMAT(1H0,23HDELTA ANGLE OF ATTACK =,F10.4,2X,4HDEG.)
 6082 FORMAT(1H0,27HWING OPTIMIZED FOR CL BAR =,F10.4)
 6084 FORMAT(1H , 18X,9HXCP BAR =,F10.4)
 6090 FORMAT(1H0,56HINCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AX
     1IS =,F10.4,2X,4HDEG.)
 6092 FORMAT(1H0,43HANGLE OF ATTACK WITH RESPECT TO BODY AXIS =,F10.4,
     12X,4HDEG.)
 6094 FORMAT(1H0,38HHEIGHT OF WING PLANE ABOVE BODY AXIS =,F10.4)
      END
```

```
      SUBROUTINE FROOTA(R,J)
C
      DIMENSION R(10),F(10)
C
C     FROOTA IS A ROOT FINDER, DESIGNED TO BE CALLED REPEATEDLY
C     UNTIL CODE J INDICATES THAT A ROOT HAS BEEN FOUND.
C
C     ASSUME A SINGLE-VALUED FUNCTION F(X) WHICH THE CALLING PROGRAM
C     CAN EVALUATE FOR ANY X.  FURTHER, THE PROGRAM KNOWS THAT THE
C     ROOT LIES BETWEEN XA AND XB.
C
C     ON THE FIRST CALL FOR A CASE, THE PROGRAM STORES
C     XA,F(XA),XB,F(XB) IN R(1),R(2),R(3),R(4), 0. IN R(9),
C     AND A TOLERANCE IN R(10).
C     FROOTA THEN COMPUTES TWO VALUES OF X, SAY XC AND XD,WHICH IT
C     STORES IN R(1) AND R(3), AND SETS J=1 WHICH INDICATES THAT
C     IT WANTS TO BE CALLED AGAIN.  THE CALLING PROGRAM
C     RESPONDS BY STORING F(XC) IN R(2) AND F(XD) IN R(4), AND
C     THEN CALLING FROOTA.THIS PROCESS CONTINUES UNTIL FROOTA
C     SIGNALS THAT THE ROOT HAS BEEN FOUND BY SETTING J=0.
C
C     FROOTA KEEPS THE POINTS WHICH BRACKET THE ROOT IN R(5)-R(8).
C     THE ROOT IS CONSIDERED TO BE FOUND WHEN
C     (1) AN EXACT ROOT IS FOUND,
C  OR (2) THE DIFFERENCE IN X-VALUES OF THE BRACKETING POINTS IS
C         SMALLER THAN THE GIVEN TOLERANCE IN R(10)
C  OR (3) THE ROOT HAS BEEN LOCATED TO THE NEAREST 1-BIT CHANGE IN X.
C
C     FROOTA ESTIMATES ONE VALUE OF X AS HALF-WAY BETWEEN THE X,S OF
C     THE BRACKETING POINTS, AND THE OTHER BY ASSUMING THE
C     FUNCTION IS LINEAR BETWEEN THE BRACKETING POINTS.
C
C     J IS SET TO -2 OR -3 IF IT DETECTS AN ERROR.
C     J = -2 IF R(9) IS .LT. 0.
C     J = -3 IF (ON THE 1ST CALL ONLY) THE POINTS IN R(1)-R(4)
C     DO NOT BRACKET THE ROOT (I.E., IF F(A)*F(B) .GT. 0.)
C
C     R(9) IS A COUNTER (INCREASED BY 1. AT EACH CALL).

      J=1                                                       ,1
      DO 100 I=1,10                                             ,2
 100  F(I)=R(I)                                                 ,3    ,4
      F(9)=F(9)+1.
C                                                               ,5
      IF(F(9)-1.)200,300,1000                                   ,6
 200  J=-2                                                      ,7
      GO TO 8000
C                                                               ,8
 300  IF(F(2)*F(4))500,1300,400                                 ,9
 400  J=-3                                                      ,10
      GO TO 8000
C     THE BRACKETING POINT WITH THE SMALLER X IS STORED IN R(5),R(6)  ,11
 500  IF(F(1) .LT. F(3)) GO TO 1800                             ,12   ,13   ,14
```

```
 600 F(5)=F(3)                                                      ,15
     F(6)=F(4)                                                      ,16
     GO TO 1700
C                                                                   ,17
1000 IF(F(1) .LE. F(3)) GO TO 1200
C    REVERSE THE 2 NEW POINTS SO ONE WITH SMALLER X WILL BE FIRST   ,18    ,19    ,20
1100 F(1)=F(3)                                                      ,21
     F(2)=F(4)                                                      ,22
     F(3)=R(1)                                                      ,23
     F(4)=R(2)
C                                                                   ,24
1200 CONTINUE                                                       ,25
     IF(F(2)*F(4)) 18C0,1300,1500
C                                                                   ,26
1300 CONTINUE                                                       ,27
     IF(F(2))1400,3000,1400                                         ,28
1400 F(1)=F(3)                                                      ,29
     GO TO 3000
C                                                                   ,30
1500 IF(F(2)*F(6)) 1700,1700,1600                                   ,31
1600 F(5)=F(3)                                                      ,32
     F(6)=F(4)                                                      ,33
     GO TO 2000
C                                                                   ,34
1700 CONTINUE                                                       ,35
     F(7)=F(1)                                                      ,36
     F(8)=F(2)                                                      ,37
     GO TO 2000
C                                                                   ,38
1800 CONTINUE                                                       ,39
     F(5)=F(1)                                                      ,40
     F(6)=F(2)                                                      ,41
     F(7)=F(3)                                                      ,42
     F(7)=F(3)                                                      ,43
     F(8)=F(4)                                                      ,44
     IF(F(1) .EQ. F(7)) GO TO 3000
C                                                                   ,45    ,46    ,47
2000 CONTINUE
C    FIND NEW TRIAL VALUES OF X                                     ,48
     F(3)=.5*(F(5)+F(7))                                            ,49
     U=F(7)-F(5)                                                    ,50
     V=F(8)-F(6)
C    WE COMPUTE F(1) IN TWO WAYS TO IMPROVE ACCURACY                ,51
     IF(ABS(F(6))-ABS(F(8))) 2100,2100,2300                         ,52
2100 F(1)=F(5)-(F(6)/V)*U                                           ,53
     IF(F(1) .EQ. F(5)) GO TO 3000                                  ,54    ,55    ,56
     GO TO 2500                                                     ,57
2300 F(1)=F(7)-(F(8)/V)*U
C                                                                   ,58
2500 CONTINUE                                                       ,59
     IF(U .GT. ABS(F(10))) GO TO 8000
C                                                                   ,60    ,61    ,62
3000 CONTINUE                                                       ,63
     J=0
```

```
     PRODTA                                                       07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C                                                                  ,64
 8000 CONTINUE                                                     ,65
      DO 8100 I=1,10                                               ,66
 8100 R(I)=F(I)                                                    ,67    ,68
 9000 RETURN                                                       ,69
      END                                                          ,70
```

```
        FRUN                                                                    05/22/67
      ASSEMBLED TEXT.

              $TEXT   FRUN                                                               FRUN0001

BINARY CARD ID. FRUN0002
        00000  000000000000     10000         LDIR
        00001  265164456060     10000
        00002  1 00000 0 00007  10001  UNLOAD SAVE    4,I
        00003  0774 00 4 00000  10000
        00004  0441 00 0 00006  10001
        00005  0020 00 4 00001  10000
        00006  0 00000 0 00000  10000
        00007  0604 00 0 00006  10001
        00010  0634 00 4 05000  10011
        00011  0634 00 4 00000  10001
        00012  0634 00 4 00003  10001
        00013  0500 60 4 00003  10000         CLA*    3,4
        00014  0601 00 0 03000  10011         STO     TEMP
        00015  000000000000     00010         CALL    .FVIO.(TEMP,TEMP+1)
        00015  0074 00 4 04000  10011
        00016  1 00002 0 01004  10011
        00017  0 00000 0 00005  10100
        00020  0 00000 0 03000  10011
        00021  0 00000 0 03001  10011

BINARY CARD ID. FRUN0003
        00022  0535 00 4 03001  10011         LAC     TEMP+1,4
        00023  0636 00 4 00025  10001         SCA     SETB,4
        00024  0074 00 4 06000  10011  RUN    TSX     .CLOSE,4
        00025  0 00000 0 00000  10000  SETB   PZE     **
                         00026         NOP    RETURN  UNLOAD
                         00027                LORG
                                       ERASE  CONTRL  ER
        00027  100000000027     00001         USE     ER
        00027  200000000002     00001  TEMP   BSS     2
        00031  200000000022     00001         BSS     18
                         00000  01111         END
```

```
          FRUN                                                                   05/22/67
        CONTROL DICTIONARY
                                                                                          FRUN0004
               $CDICT FRUN

BINARY CARD ID. FRUN0005
                000053000000                PREFACE     START=0,LENGTH=43,TYPE=7094,CMPLX=5
                000004000005
                265164456060       FRUN     DECK        LOC=0,LENGTH=43
                000053000000
                644543462124       UNLOAD   REAL        LOC=2,LENGTH=0
                000000000002
                255121622560       ERASE    REAL        LOC=27,LENGTH=20
                000024000027
                332665314633       .FVIO.   VIRTUAL     SECT. 4,CALL
                200000100000
                627062434623       SYSLOC   VIRTUAL     SECT. 5
                200000000000
                332343466225       .CLOSE   VIRTUAL     SECT. 6
                200000000000
                                                                                          FRUN0006
               $DKEND FRUN


NO MESSAGES FOR THIS ASSEMBLY
```

```
                                             *     REWRITTEN FOR IBMAP / 3-3-63 / E. WRIGHT - TRANSPORT -BOEING      FTAN0020
                                             *     METHODS MEMO 7090-43                                              FTAN0030
                                             *                                                                       FTAN0040
                                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0050
                                             *                                                                       FTAN0060
                                             *     A=TAN (B)                                                         FTAN0070
                                             *                                                                       FTAN0080
                                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0090
                                                                                                                     FTAN0100
                                             *      ENTRY   TAN                                                      FTAN0110
                                                                                                                     FTAN0120
                                                                                                                     FTAN0130
                                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN0140
BINARY CARD (NOT PUNCHED)
       00000  1 00000 0 00004   10001   TAN     SAVE     1                                                           FTAN0150
       00001  0774 00 1 00000   10000
       00002  0774 00 4 00000   10000
       00003  0020 00 4 00001   10000
       00004  0634 00 4 05000   10011
       00005  0634 00 4 00162   10001
       00006  0634 00 4 00002   10001
       00007  0634 00 1 00001   10001
       00010  0500 60 4 00003   10000           CLA*     3,4                                                         FTAN0160
       00011  0601 00 0 03000   10011           STO      COMMON         X                                            FTAN0170
       00012  0760 00 0 00003   10000           SSP                     ABS X                                        FTAN0180
       00013  0560 00 0 00164   10001           LDQ     =210828713.     TEST IF X GREATER THAN PI*(2**26)            FTAN0190
       00014  0040 00 0 00116   10001           TLQ      ERROR1         YES, ERROR                                   FTAN0200
       00015  0560 00 0 00165   10001           LDQ     =3.14159265     NO,  TEST IF X GREATER THAN PI               FTAN0210
       00016  0040 00 0 00023   10001           TLQ      TAN1           YES, NEEDS REDUCING                          FTAN0220
       00017  0560 00 0 00166   10001           LDQ     =.0001          NO,  TEST IF X GREATER THAN .0001            FTAN0230
       00020  0040 00 0 00034   10001           TLQ      TAN2           YES, X IN PROPER RANGE                       FTAN0240
       00021  0500 00 0 03000   10011           CLA      COMMON         NO                                           FTAN0250
                        00022                   RETURN   TAN            RETURN X = TAN X                             FTAN0260
                                        *                                                                            FTAN0270
                                        * REDUCE X TO MODULO PI                                                      FTAN0280
                                        *                                                                            FTAN0290
BINARY CARD (NOT PUNCHED)
       00023  4500 00 0 03000   10011   TAN1    CAL      COMMON         X GREATER THAN PI (N.M*PI)                   FTAN0300
       00024  0630 00 0 00167   10001           STP     =67108864.      MAKE SIGN SAME AS X                          FTAN0310
       00025  4130 00 0 00000   10000           XCL                                                                  FTAN0320
       00026  0260 00 0 00170   10001           FMP     =.318309886     1/PI (LEAVES N.M)                            FTAN0330
       00027  0300 00 0 00167   10001           FAD     =67108864.      2**26 (SEPARATES N AND M)                    FTAN0340
       00030  0131 00 0 00000   10000           XCA                     KEEP FRACTIONAL PART (.M)                    FTAN0350
       00031  0300 00 0 00171   10001           FAD     =17179869184    NORMALIZE                                    FTAN0360
       00032  0241 00 0 00170   10001           FDP     =.318309886     1/PI (LEAVES .M*PI)                          FTAN0370
       00033  4600 00 0 03000   10011           STQ      COMMON         X REPLACED BY X MODULO PI                    FTAN0380
                                        *                                                                            FTAN0390
                                        * TEST IF X IS A MULTIPLE  OF PI/2                                           FTAN0400
                                        *                                                                            FTAN0410
       00034  0500 00 0 00172   10001   TAN2    CLA     =1.57079633     PI/2                                         FTAN0420
       00035  4400 00 0 03000   10011           SBM      COMMON                                                      FTAN0430
```

```
        FTAN                                                              07/31/65
        IBJOB LIBRARY / TANGENT FUNCTION

    00036  0771 00 0 00001  10000       ARS     1             ALLOW 1 BIT RANGE                      FTAN0440
    00037  0100 00 0 00124  10001       TZE     ERROR2        ERROR, X=PI/2                          FTAN0450
                                  *                                                                  FTAN0460
                                  * IF X GREATER THAN PI/2, PLACE IN PROPER QUADRANT                 FTAN0470
                                  *                                                                  FTAN0480
    00040  0500 00 0 03000  10011       CLA     COMMON        X                                      FTAN0490
    00041  4130 00 0 00000  10000       XCL                   ABS X                                  FTAN0500
    00042  0500 00 0 00172  10001       CLA     =1.57079633   TEST IF X GREATER THAN PI/2            FTAN0510
    00043  0040 00 0 00053  10001       TLQ     TAN4          NO,  X OK                              FTAN0520
    00044  0500 00 0 03000  10011       CLA     COMMON        YES, PLACE IN PROPER QUADRANT          FTAN0530
    00045  4120 00 0 00051  10001       TMI     TAN3                                                 FTAN0540

BINARY CARD (NOT PUNCHED)
    00046  0302 00 0 00165  10001       FSB     =3.14159265   QUADRANT 2                             FTAN0550
    00047  0601 00 0 03000  10011       STO     COMMON        X REPLACED BY X-PI                     FTAN0560
    00050  0020 00 0 00054  10001       TRA     TAN5                                                 FTAN0570
    00051  0300 00 0 00165  10001  TAN3 FAD     =3.14159265   QUADRANT 3                             FTAN0580
    00052  0601 00 0 03000  10011       STO     COMMON        X REPLACED BY -X+PI                    FTAN0590
                                  *                                                                  FTAN0600
                                  *                                                                  FTAN0610
    00053  0500 00 0 03000  10011  TAN4 CLA     COMMON        X                                      FTAN0620
    00054  0760 00 0 00003  10000  TAN5 SSP                   ABS X                                  FTAN0630
    00055  0560 00 0 00173  10001       LDQ     =0.78539816   TEST IF X GREATER THAN PI/4            FTAN0640
    00056  0040 00 0 00061  10001       TLQ     TAN6          YES                                    FTAN0650
                                  *                                                                  FTAN0660
                                  * TAN X = CONTINUED FRACTION                                       FTAN0670
                                  *                                                                  FTAN0680
    00057  0074 00 4 00076  10001       TSX     CONTFR,4      NO                                     FTAN0690
    00060  0020 00 0 00074  10001       TRA     TAN9                                                 FTAN0700
                                  *                                                                  FTAN0710
                                  * TAN X = 1/TAN(PI/2-ABS X)                                        FTAN0720
                                  *                                                                  FTAN0730
    00061  0500 00 0 03000  10011  TAN6 CLA     COMMON        IS X NEGATIVE                          FTAN0740
    00062  4120 00 0 00066  10001       TMI     TAN7          YES                                    FTAN0750
    00063  0302 00 0 00172  10001       FSB     =1.57079633   NO                                     FTAN0760
    00064  0760 00 0 00002  10000       CHS                   REPLACE X BY PI/2-X                    FTAN0770
    00065  0020 00 0 00070  10001       TRA     TAN8                                                 FTAN0780
    00066  0300 00 0 00172  10001  TAN7 FAD     =1.57079633                                          FTAN0790
    00067  0760 00 0 00002  10000       CHS                   REPLACE X BY -(PI/2-ABS X)             FTAN0800
    00070  0601 00 0 03000  10011  TAN8 STO     COMMON                                               FTAN0810

BINARY CARD (NOT PUNCHED)
    00071  0074 00 4 00076  10001       TSX     CONTFR,4      TAN(PI/2-ABS X)                        FTAN0820
    00072  0500 00 0 00161  10001       CLA     A1            1.0                                    FTAN0830
    00073  0241 00 0 03002  10011       FDP     COMMON+2      TAN X = 1/TAN(PI/2-ABS X)              FTAN0840
                                  *                                                                  FTAN0850
    00074  0131 00 0 00000  10000  TAN9 XCA                   ANSWER IN ACCUMULATOR                  FTAN0860
                      00075             RETURN  TAN           RETURN                                 FTAN0870
                                  *                                                                  FTAN0880
                                  * TAN X = X/(1-X**2/(3-X**2/(5-X**2/(7-X**2/(9-X**2/11)))))        FTAN0890
                                  *                                                                  FTAN0900
    00076  0774 00 1 00005  10000  CONTFR AXT   5,1                                                  FTAN0910
    00077  0560 00 0 03000  10011       LDQ     COMMON        X                                      FTAN0920
    00100  0260 00 0 03000  10011       FMP     COMMON                                               FTAN0930
```

```
00101  0601 00 0 03001  10011        STO    COMMON+1     X**2                                      FTAN0940
00102  0500 00 0 00154  10001        CLA    A6           A(6)=11.                                  FTAN0950
00103  0601 00 0 03002  10011        STO    COMMON+2                                               FTAN0960
00104  0502 00 0 03001  10011  TAN12 CLS    COMMON+1     -X**2                                     FTAN0970
00105  0241 00 0 03002  10011        FDP    COMMON+2                                               FTAN0980
00106  0131 00 0 00000  10000        XCA                                                           FTAN0990
00107  0300 00 1 00162  10001        FAD    A1+1,1       A(I)                                      FTAN1000
00110  0601 00 0 03002  10011        STO    COMMON+2                                               FTAN1010
00111  2 00001 1 00104  10001        TIX    TAN12,1,1    5 TIMES                                   FTAN1020
00112  0500 00 0 03000  10011        CLA    COMMON       X                                         FTAN1030
00113  0241 00 0 03002  10011        FDP    COMMON+2                                               FTAN1040

BINARY CARD (NOT PUNCHED)
00114  4600 00 0 03002  10011        STO    COMMON+2                                               FTAN1050
00115  0020 00 4 00001  10000        TRA    1,4                                                    FTAN1060
                               *                                                                   FTAN1070
                               * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1080
                               *                                                                   FTAN1090
00116  000000000000     00010  ERROR1 CALL  .FXEM.(BAIL)  ABS X TOO LARGE                          FTAN1100
00116  0074 00 4 04000  10011
00117  1 00001 0 01003  10011
00120  0 00162 0 00202  10100
00121  0 00000 0 00134  10001
00122  4754 00 0 00000  10000        ZAC                 SET RESULT TO ZERO                        FTAN1110
                 00123               RETURN TAN                                                    FTAN1120
00124  000000000000     00010  ERROR2 CALL  .FXEM.(GIVE)  X ODD MULTIPLE                           FTAN1130
00124  0074 00 4 04000  10011
00125  1 00001 0 01003  10011
00126  0 00162 0 00217  10100
00127  0 00000 0 00144  10001
00130  0500 00 0 03000  10011        CLA    COMMON       SET RESULT TO LARGEST NUMBER              FTAN1140
00131  0760 00 0 00000  10000        CLM                   WITH CORRECT SIGN.                      FTAN1150
00132  0760 00 0 00006  10000        COM                                                           FTAN1160
                 00133               RETURN TAN                                                    FTAN1170
00134  000000000103     10000  BAIL  DEC    67                                                     FTAN1171

BINARY CARD (NOT PUNCHED)
00135  0 00003 0 00137  10001        PZE    MESA,,3                                                FTAN1172
00136  0 00002 0 00142  10001        PZE    MESB,,2                                                FTAN1173
00137  002122626067     10000  MESA  BCI    3,0ABS X TOO LARGE                                     FTAN1174
00140  606346466043     10000
00141  215127256060     10000
00142  002145626013     10000  MESB  BCI    2,0ANS = 0                                             FTAN1175
00143  600060606060     10000
00144  000000000102     10000  GIVE  DEC    66                                                     FTAN1176
00145  0 00003 0 00147  10001        PZE    MESC,,3                                                FTAN1177
00146  0 00002 0 00152  10001        PZE    MESD,,2                                                FTAN1178
00147  006760462424     10000  MESC  BCI    3,0X ODD MULTIPLE                                      FTAN1179
00150  604464436331     10000
00151  474325606060     10000
00152  002145621333     10000  MESD  BCI    2,0ANS=.17E39                                          FTAN117A
00153  010725031160     10000
                               *                                                                   FTAN1180
                               * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1190
```

```
                                   *                                                       FTAN1200
      00154  204540000000   10000   A6     DEC     11.,9.,7.,5.,3.                         FTAN1210
      00155  204440000000   10000
      00156  203700000000   10000
      00157  203500000000   10000

  ARY CARD (NOT PUNCHED)
      00160  202600000000   10000
      00161  201400000000   10000   A1     DEC     1.                                      FTAN1220
                                   *                                                       FTAN1230
                                   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1240
                                   *                                                       FTAN1250
      00162  000000000000   10000   LK.DR LDIR                                             FTAN1260
      00163  266321456060   10000
      00164  234622077325   10000          LORG                                            FTAN1270
      00165  202622077325   10000
      00166  163643334273   10000
      00167  233400000000   10000
      00170  177505746033   10000
      00171  200000000000   10000
      00172  201622077325   10000
      00173  200622077325   10000
                                   *                                                       FTAN1280
                                   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1290
                                   *                                                       FTAN1300
                                    ERASE CONTRL   ER                                      FTAN1310
      00174  100000000174   00001          USE     ER                                      FTAN1320
      00174  200000000003   00001  COMMON BSS      3                                       FTAN1330
      00177  200000000021   00001     .    BSS     17                                      FTAN1340
                                   *                                                       FTAN1350
                                   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * FTAN1360
                                                                                           FTAN1370
                    00000   01111          END                                             FTAN1380
```

```
                                                                                      3FA00030
                                                 CALLING SEQUENCE IS CALL .FVIO.(LN,ERAS) WHERE LN IS      3FA00040
                                                 LOCATION OF VARIABLE LOGICAL UNIT AND ERAS WILL CONTAIN   3FA00050
                                                 CONTENTS OF APPROPRIATE .UNXX.  (.UNXX. CONTAINS          3FA00060
                                                 PZE UNITXX WHERE XX CORRESPONDS TO LOGICAL UNIT N).       3FA00070
                                                 FVIO IS CALLED FOR ANY I/O STATEMENT SPECIFYING           3FA00080
                                                 A VARIABLE LOGICAL UNIT.                                  3FA00090
                                                                                                           3FA00100

BINARY CARD (NOT PUNCHED)
       00000  1 00000 0 00004  10001  .FVIO. SAVE    (2)                                          (1)3FA00110
       00001  0774 00 2 00000  10000
       00002  0774 00 4 00000  10000
       00003  0020 00 4 00001  10000
       00004  0634 00 4 04000  10011
       00005  0634 00 4 00102  10001
       00006  0634 00 4 00002  10001
       00007  0634 00 2 00001  10001
       00010  0500 60 4 00003  10000         CLA*    3,4                 PICK UP LOGICAL UNIT NUMBER   (1)3FA00120
       00011  0737 00 2 00000  10000         PAC     ,2                                                (1)3FA00130
       00012  7 00000 2 00016  11101         TXL     ERROR,2,-NUNITS-1   IS UNIT ZERO, OR TOO LARGE    (1)3FA00140
              1 00070 0 00102  10101
              1 00001 0 00000  10011
              1 00000 0 00000  11100
              1 00001 7 00000  10011
       00013  0500 60 2 00070  10001         CLA*    TABLE,2             SAVE ADDRESS OF               (1)3FA00150
       00014  0601 60 4 00004  10000         STO*    4,4                 FILE CONTROL BLOCK            (1)3FA00160
                        00015                RETURN  .FVIO.                                            (1)3FA00170
       00016  0500 60 4 00003  10000  ERROR  CLA*    3,4                 LOGICAL UNIT IN ERROR         (1)3FA00250

BINARY CARD (NOT PUNCHED)
       00017  4320 00 0 00064  10001         ANA     ADMSK               DEFINED FOR THIS UNIT VALUE.     3FA00260
       00020  0131 00 0 00000  10000         XCA                         CONVERT THIS ILLEGAL VALUE       3FA00270
       00021  0774 00 4 00000  10000         AXT     0,4                 TO DECIMAL FOR ERROR MESSAGE.    3FA00280
       00022  0600 00 0 00065  10001         STZ     TEMP                                                 3FA00290
       00023  0754 00 0 00000  10000  CNVT   PXA     0,0                                                  3FA00300
       00024  0221 00 0 00066  10001         DVP     L(10)                                                3FA00310
       00025  0767 00 4 00000  10000         ALS     0,4                                                  3FA00320
       00026  4602 00 0 00065  10001         ORS     TEMP                                                 3FA00330
       00027  0500 00 0 00104  10001         CLA     =1                                                   3FA00340
       00030  0040 00 0 01002  10011         TLQ     *+2                                                  3FA00350
       00031  1 77772 4 00023  10001         TXI     CNVT,4,-6                                            3FA00360
       00032  4500 00 0 00067  10001         CAL     BLANKS                                               3FA00370
       00033  0767 00 4 00006  10000         ALS     6,4                                                  3FA00380
       00034  4501 00 0 00065  10001         ORA     TEMP                                                 3FA00390
       00035  0602 00 0 00054  10001         SLW     E47MES+6                                             3FA00400
       00036  000000000000     00010  UNERR  CALL    .FXEM.(CODE)        EXIT FOR EXECUTION ERROR.        3FA00410
       00036  0074 00 4 03000  10011
       00037  1 00001 0 01003  10011
       00040  0 00102 0 00053  10100

BINARY CARD (NOT PUNCHED)
       00041  0 00000 0 00043  10001
```

```
       00042  0020 00 0 05000  10011         TRA     .LXERR              NO OPTIONAL EXIT.                3FA00420
       00043  0 00000 0 00057  10000  CODE   PZE     47                                                   3FA00430
       00044  0 00007 0 00046  10001  ER47MS PZE     E47MES,,7                                            3FA00440
       00045  0 00007 0 00055  10001  ER47OP PZE     NOOPXT,,7                                            3FA00450
       00046  004346273123     10000  E47MES BCI     7,0LOGICAL UNIT NOT DEFINED FOR VALUE                3FA00460
       00047  214360644531     10000
       00050  636045466360     10000
       00051  242526314525     10000
       00052  246026465160     10000
       00053  652143642560     10000
       00054  606060606060     10000
       00055  004546604647     10000  NOOPXT BCI     7,0NO OPTIONAL EXIT - EXECUTION TERMINATED           3FA00470
       00056  633146452143     10000
       00057  602567316360     10000
       00060  406025672523     10000
       00061  646331464560     10000
       00062  632551443145     10000
       00063  216325246060     10000

BINARY CARD (NOT PUNCHED)
       00064  000000077777     10000  ADMSK  OCT     000000077777                                         3FA00480
       00065  0 00000 0 00000  10000  TEMP   PZE     **                                                   3FA00490
       00066  000000000012     10000  L(10)  DEC     10                                                   3FA00500
       00067  606060606060     10000  BLANKS BCI     1,                                                   3FA00510
       00070  0 00000 0 00000  10011  TABLE  PZE     NUNITS                                               3FA00550
              1 00070 0 00102  10101
              1 00001 7 00000  10011
       00071  0 00000 0 06000  10011         PZE     .UN01.                                               3FA00560
       00072  0 00000 0 07000  10011         PZE     .UN02.                                               3FA00570
       00073  0 00000 0 10000  10011         PZE     .UN03.                                               3FA00580
       00074  0 00000 0 11000  10011         PZE     .UN04.                                               3FA00590
       00075  0 00000 0 12000  10011         PZE     .UN05.                                               3FA00600
       00076  0 00000 0 13000  10011         PZE     .UN06.                                               3FA00610
       00077  0 00000 0 14000  10011         PZE     .UN07.                                               3FA00620
       00100  0 00000 0 15000  10011         PZE     .UN08.                                               3FA00630
                                      *      ADDITIONAL UNITS MAY BE INSERTED HERE. FOR EACH UNIT         3FA00640
                                      *      INSERTED, A CORRESPONDING ROUTINE MUST BE INSERTED TO        3FA00650
                                      *      PRODUCE 4 $FILE CARD FOR THE ADDITIONAL UNIT.                3FA00650
       00101  0 00000 0 16000  10011         PZE     .UN09.
                        00011         NUNITS EQU     *-TABLE-1                                            3FA00670
       00102  000000000000     10000         *LDIR
       00103  266531466060     10000
       00104  000000000001     10000         *LORG

BINARY CARD (NOT PUNCHED)
                        00000  01111         END                                                          3FA00680
```

```
      SUBROUTINE GEOMD(DAT,LI,LC,LERR,LTAPE)
C     GEOMETRIC DEFINITION LINK
      DIMENSION BUF(10000),DICT(9),NU(3),LOK(3,9),DAT(2)
     1 ,BAXIS(2),BTITLE(12),WTITLE(12)
      COMMON  /LGEOMD/LGDEF(3,6)
      LOGICAL LGDEF,LGTAPE,LTDUMP
      EQUIVALENCE  (LOK(1,1),LUWPL),(LOK(2,1),MUWPL),(LOK(3,1),NUWPL),
     X(LOK(1,2),LLWPL),(LOK(2,2),MLWPL),(LOK(3,2),NLWPL),
     X(LOK(1,3),LUWPLX),(LOK(2,3),MUWPLX),(LOK(3,3),NUWPLX),
     X(LOK(1,4),LLWPLX),(LOK(2,4),MLWPLX),(LOK(3,4),NLWPLX),
     X(LOK(1,5),LBPL),(LOK(2,5),MBPL),(LOK(3,5),NBPL),
     X(LOK(1,6),LSTA),(LOK(3,6),NSTA),
     X(LOK(1,7),LWPF),(LOK(2,7),MWPF),(LOK(3,7),NWPF),
     X(LOK(1,8),LUWB),(LOK(3,8),NUWB),
     X(LOK(1,9),LLWB),(LOK(3,9),NLWB)
      DATA  DICT/4HBODY,4HBODY,4HWING,4HWING,6HDEFEND,3HWBX,
     1      6HNOTAPE,6HNOTAPE,5HTDUMP/,LGTAPE/T/,
     1      NBUF,NDICT,NU(3),LOK/10000, 9,50,27*0/
C
C     INTWB IS USED BY DEFEND
      INTWB=0                                                        ,1
      LTDUMP=.FALSE.
C
C     THE LGDEF ARRAY (LOGICAL) TELLS THE STATUS OF DEFINITION.
C     LGDEF(I,J) HAS FOLLOWING FORMAT.
C     J=1, UPPER WING.  J=2, LOWER WING.  J=3, BODY.  J=4, WING
C     PLANFORM.  J=5, UPPER WING INTERSECTION.  J=6, LOWER WING.
C     I=1, REQUESTED.  I=2, SUCCESSFULLY DEFINED.
C     I=3, DEFINITION ON TAPE (LTAPE).
C                                                                    ,2
      DO 100 J=1,6                                                   ,3
      DO 100 I=1,3                                                   ,4
  100 LGDEF(I,J)=.FALSE.
C
C     CHECK LTAPE                                                    ,5    ,6    ,7
      IF (LTAPE*(LTAPE-40))430,420,420                               ,8
 420  LGTAPE=.FALSE.
C                                                                    ,9
 430  NU(2)=LERR
C     SET UP STORAGE BUFFER                                          ,10
      CALL INIBFR(BUF,NBUF)
C
C     READ DATA CARD, SEE IF VALID COMMAND                           ,11
  500 WRITE(LO,501)                                                  ,12   ,13
  501 FORMAT(1H1)
 502  IGO=INTURP(DICT,NDICT,LI,LO)
C
C                                                                    ,14
      IF (IGO)510,510,530
C     NOT VALID                                                      ,15
 510  WRITE (LO,520)                                                 ,16   ,17
 520  FORMAT(1H+  90X,7HCOMMENT)
      GO TO 502
```

```
        GEOMD                                                               10/04/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)
C
  530 GO TO (800,800,1000,1000,1600,2000,1700,1700,                          ,18
     X 2100),IGO
C
C     BODY
  800 CALL BODY1(DAT,LI,LO,BUF,LSTA,NSTA,LBPL,MBPL,NBPL,LTHETA               ,19
     1 ,BAXIS,BTITLE,LGDEF,NU)                                               ,20
  810 GO TO 500
C
C     WING                                                                   ,21
 1000 CALL WINGA(BUF,LI,LO,DAT,LGDEF,LOK,LWPF,WTITLE,NU)                     ,22
 1010 CONTINUE                                                               ,23
      GO TO 500
C
C     DEFEND                                                                 ,24
 1600 GO TO 2050
C
C     NOTAPE                                                                 ,25
 1700 LGTAPE=.FALSE.                                                         ,26
      GO TO 500
C
C     WBX                                                                    ,27
 2000 INTWB=-1
C                                                                            ,28
 2050 CALL DEFEND(BUF,LOK,LO,LTAPE,LGDEF,LGTAPE,INTWB)                       ,29
      IF(INTWB)2060,2080,2080
C                                                                            ,30
 2060 INTWB=1                                                                ,31
      CALL WBXUL(BUF,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU)             ,32
 2070 GO TO 500
C                                                                            ,33
 2080 IF(.NOT. LTDUMP) GO TO 9000                                            ,34   ,35   ,36
      CALL TDUMP(LO,LTAPE,LGDEF)                                             ,37
      GO TO 9000
C
C     TDUMP                                                                  ,38
 2100 LTDUMP=.TRUE.                                                          ,39
      GO TO 500
C                                                                            ,40
 9000 CALL CLEAR(BUF)                                                        ,41
      RETURN                                                                 ,42
      END                                                                    ,43
```

```
      SUBROUTINE GRIDS
      COMMON /GRIDA/ MODE,DX,DY,DZ,IX,IY,IZ,XO,YO,ZO,XA,YA,ZA,LIT
      COMMON DUM(4),N8,DM(3),N5,N6
      COMMON /INTGA/GP(3,500),VLS(4,500)
      DATA BX/1HX/,BY/1HY/,BZ/1HZ/
      DATA NMAX/500/
      INTEGER BX,BY,BZ
 1000 FORMAT(1H0,3F15.6,5X,3F15.7,5X,F15.7)
 1001 FORMAT(1H0,24HY-Z CUT, FOR THIS CUT X=F10.5)
 1002 FORMAT(1H0,7X,1HX,14X,1HY,14X,1HZ,16X,1HU,14X,1HV,14X,1HW,16X,2HCP
     $)
 1003 FORMAT(1H0,24HX-Z CUT, FOR THIS CUT Y=F10.5)
 1004 FORMAT(1H0,24HX-Y CUT, FOR THIS CUT Z=F10.5)
 1005 FORMAT(1H-,35HOPTION REQUESTED---RECTANGULAR GRID)
 1006 FORMAT(1H-,28HOPTION REQUESTED---SKEW GRID)
 1007 FORMAT(1H1)
      IF(BLUNDR(IX,IY,IZ,MODE,NMAX).NE.0.) RETURN                        2
      IF(MODE.EQ.4) GO TO 101
      WRITE(N6,1005)                                                     7
      GO TO (1,21,40),MODE
    1 DO 20 I=1,IX
      IC=0
      DO 5  K=1,IZ
      DO 5  J=1,IY
      IC=IC+1
      GP(1,IC)=XO+FLOAT(I-1)*DX
      GP(2,IC)=YO+FLOAT(J-1)*DY
    5 GP(3,IC)=ZO+FLOAT(K-1)*DZ
      CALL COMP(GP,VLS,IC)                                              25
      WRITE(N6,1007)                                                    26
      WRITE(N6,1001) GP(1,1)                                            27
      WRITE(N6,1002)                                                    28
      WRITE(N6,1000) (GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J),
     $,VLS(4,J),J=1,IC)                                                 29
      CALL FLOOUT(IC)                                                   40
   20 CONTINUE
      GO TO 500
   21 CONTINUE
      DO 30 J=1,IY
      IC=0
      DO 22 K=1,IZ
      DO 22 I=1,IX
      IC=IC+1
      GP(1,IC)=XO+FLOAT(I-1)*DX
      GP(2,IC)=YO+FLOAT(J-1)*DY
   22 GP(3,IC)=ZO+FLOAT(K-1)*DZ
      CALL COMP(GP,VLS,IC)                                              61
      WRITE(N6,1007)                                                    62
      WRITE(N6,1003) GP(2,1)                                            63
      WRITE(N6,1002)                                                    64
      WRITE(N6,1000) (GP(1,L),GP(2,L),GP(3,L),VLS(1,L),VLS(2,L),VLS(3,L)
     $,VLS(4,L),L=1,IC)                                                 65
      CALL FLOOUT(IC)                                                   76
```

```
 30 CONTINUE
    GO TO 500
 40 CONTINUE
    DO 50 K=1,IZ
    IC=0
    DO 42 I=1,IX
    DO 42 J=1,IY
    IC=IC+1
    GP(1,IC)=XO+FLOAT(I-1)*DX
    GP(2,IC)=YO+FLOAT(J-1)*DY
 42 GP(3,IC)=ZO+FLOAT(K-1)*DZ
    CALL COMP(GP,VLS,IC)                                                   97
    WRITE(N6,1004) GP(3,1)                                                 98
    WRITE(N6,1002)                                                         99
    WRITE(N6,1000) (GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)
   $,VLS(4,J),J=1,IC)                                                      100
    CALL FLOOUT(IC)                                                        111
 50 CONTINUE
    GO TO 500
101 CONTINUE
    WRITE(N6,1007)                                                         116
    WRITE(N6,1006)                                                         117
    S=SQRT((XO-XA)**2+(YO-YA)**2+(ZC-ZA)**2)                               118
    XAS=S/(XA-XC)
    YAS=S/(YA-YC)
    ZAS=S/(ZA-ZO)
    IF(LIT.EQ.BY)  GO TO 120
    IF(LIT.EQ.BZ)  GO TO 130
    IF(DY.EQ.0.) GO TO 108
    DS=DY*YAS
    IS=IY
    GO TO 110
108 DS=DZ*ZAS
    IS=IZ
110 IC=0
    DO 112 I=1,IX
    DO 112 J=1,IS
    IC=IC+1
    GP(1,IC)=XO+FLOAT(J-1)*DS/XAS+FLOAT(I-1)*DX
    GP(2,IC)=YO+FLOAT(J-1)*DS/YAS
112 GP(3,IC)=ZO+FLOAT(J-1)*DS/ZAS
    CALL COMP(GP,VLS,IC)                                                   145
    WRITE(N6,1002)                                                         146
    WRITE(N6,1000) (GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)
   $,VLS(4,J),J=1,IC)                                                      147
    CALL FLOOUT(IC)                                                        160
    GO TO 500
120 CONTINUE
    IF(DX.EQ.0.) GO TO 124
    DS=DX*XAS
    IS=IX
    GO TO 125
124 DS=DZ*ZAS
    IS=IZ
```

```
125 IC=0
    DO 127 I=1,IY
    DO 127 J=1,IS
    IC=IC+1
    GP(1,IC)=XO+FLOAT(J-1)*DS/XAS
    GP(2,IC)=YO+FLOAT(J-1)*DS/YAS+FLOAT(I-1)*DY
127 GP(3,IC)=ZO+FLOAT(J-1)*DS/ZAS
    CALL COMP(GP,VLS,IC)                                                        183
    WRITE(N6,1002)                                                              184
    WRITE(N6,1000) (GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)
   $,VLS(4,J),J=1,IC)                                                           185
    CALL FLCOUT(IC)                                                             198
    GO TO 500
130 CONTINUE
    IF(DX.EQ.0.) GO TO 133
    DS=DX*XAS
    IS=IX
    GO TO 134
133 DS=DY*YAS
    IS=IY
134 CONTINUE
    IC=0
    DO 135 I=1,IZ
    DO 135 J=1,IS
    IC=IC+1
    GP(1,IC)=XO+FLOAT(J-1)*DS/XAS
    GP(2,IC)=YO+FLOAT(J-1)*DS/YAS
135 GP(3,IC)=ZO+FLOAT(J-1)*DS/ZAS+FLOAT(I-1)*DZ
    CALL COMP(GP,VLS,IC)                                                        221
    WRITE(N6,1002)                                                              222
    WRITE(N6,1000) (GP(1,J),GP(2,J),GP(3,J),VLS(1,J),VLS(2,J),VLS(3,J)
   $,VLS(4,J),J=1,IC)                                                           223
    CALL FLCOUT(IC)                                                             236
500 CONTINUE
    RETURN
    END
```

```
CIADARY     MICHAEL SYNGE            6-9247       COMPILED  21 AUG 64
      FUNCTION  IADARY(ARRAY,IN,BFR,NARAY)                          IADA0030
      DIMENSION ARRAY(1),BFR(1)                                     IADA0040
C                                                                   IADA0060
C  TO ADD AN ARRAY TO A BUFFER                                      IADA0070
C                                                                   IADA0090
C              I      - THE LENGTH OF THE CURRENT ARRAY             IADA0100
C              M      - FIRST WORD OF CURRENT ARRAY                 IADA0110
C              MM     - THE TOTAL LENGTH OF THE BUFFER              IADA0120
C              N      - THE LENGTH OF THE ARRAY TO BE ADDED         IADA0130
C                                                                   IADA0140
C              SEARCH FOR A FREE ARRAY WHICH IS BIG ENOUGH          IADA0150
C                                                                   IADA0160
      N=IABS(IN)                                                    IADA0170   ,1
      K=0                                                           IADA0180   ,2
      IF(ICODE(BFR)-4)95,1,95                                       IADA0190   ,3
    1 MM=-IBFR(BFR(3))                                              IADA0200   ,4
    2 M=4                                                           IADA0210   ,5
    3 I=LENG(BFR(M-1))                                              IADA0220   ,6
      IF( ICODE(BFR(M-2)))5,4,5                                     IADA0230   ,7
    4 IF(I-N)5,10,10                                                IADA0240   ,8
    5 M=M+I+2                                                       IADA0250   ,9
      IF(M-MM)3,6,6                                                 IADA0260
C                                                                   IADA0270
C              PACK THE BUFFER IF NECESSARY                         IADA0280
C                                                                   IADA0290   ,10
    6 IF(K)90,7,90                                                  IADA0300   ,11
    7 K=1                                                           IADA0310   ,12
      IF(IPACK(BFR))85,85,2                                         IADA0320
C                                                                   IADA0330
C              FREE ARRAY FOUND                                     IADA0340
C              SET UP CODE WORDS                                    IADA0350
C                                                                   IADA0360   ,13
   10 IADARY=M                                                      IADA0370   ,14
      IF(IN)25,25,15                                                IADA0380   ,15
   15 DO 20 K=1,N                                                   IADA0390   ,16
      BFR(M)=ARRAY(K)                                               IADA0400   ,17
 20   M=M+1                                                         IADA0410   ,18    ,19
   25 BFR(IADARY-1)=ARSEC(I,NARAY)                                  IADA0420   ,20
      BFR(IADARY-2)=ARFUL(IADARY)                                   IADA0430   ,21
   30 NARAY=IADARY                                                  IADA0440   ,22
      CALL IRLEAS(BFR(IADARY),N)                                    IADA0450   ,23
 80   RETURN                                                        IADA0460
C                                                                   IADA0470
C              ERRORS                                               IADA0480
C                                                                   IADA0490   ,24
   85 IADARY=-1                                                     IADA0500   ,25
      GO TO 80                                                      IADA0510   ,26
   90 IADARY=-2                                                     IADA0520   ,27
      GO TO 80                                                      IADA0530   ,28
   95 IADARY=-3                                                     IADA0540   ,29
      GO TO 80                                                      IADA0550   ,30
      END                                                           IADA0560   ,31
```

```
  IBFR                                                        07/31/65
 7094 RELMOD ASSEMBLY.

        $IBLDR IBFR                                                                    IBFR0000
ASSEMBLED TEXT.

        $TEXT  IBFR                                                                    IBFR0001

                                           *IBFR       MICHAEL SYNGE          6-9247      ASSEMBLED  21 AUG 64
                                           *
                                           *  TO LOCATE THE FIRST CELL OF A BUFFER
                                           *  RELATIVE TO THE FIRST CELL OF AN ARRAY
                                           *
                                           *           CALLING SEQUENCE
                                           *
                                           *                   M=IBFR(ARRAY)
                                           *
                                           *           NOTE - M IS ALWAYS NEGATIVE
                                           *           UNLESS THE GIVEN ARGUMENT IS
                                           *           NOT THE FIRST WORD OF AN ARRAY.
                                           *
                                                 ENTRY   IBFR
                                           *
BINARY CARD (NOT PUNCHED)
       00000  0500 00 4 00003  10000   IBFR  CLA     3,4
       00001  0402 00 0 00017  10001         SUB     =2
       00002  0621 00 0 00003  10001         STA     A
       00003  4500 00 0 00000  10000   A     CAL     **
       00004  0621 00 0 00016  10001         STA     B
       00005  4320 00 0 00020  10001         ANA     =077777700000
       00006  4340 00 0 00021  10001         LAS     =0700000
       00007  0020 00 0 00011  10001         TRA     ERROR
       00010  0020 00 0 00013  10001         TRA     SUCCES
       00011  0500 00 0 00022  10001   ERROR CLA     =1
       00012  0020 00 0 00015  10001         TRA     EXIT
       00013  0500 00 0 00016  10001  SUCCES CLA     B
       00014  4760 00 0 00003  10000         SSM
       00015  0020 00 4 00001  10000   EXIT  TRA     1,4
                                       *
       00016  0 00000 0 00000  10000   B
       00017  000000000002     10000         *LORG
       00020  077777700000     10000
       00021  000000700000     10000
       00022  000000000001     10000

BINARY CARD (NOT PUNCHED)
                        00000  01111         END
```

```
              $CDICT IBFR                                                        IBFR0002

BINARY CARD (NOT PUNCHED)
          000023000000              PREFACE    START=0,LENGTH=19,TYPE=7094,CMPLX=5
          000004000005
          312226516060      IBFR    DECK       LOC=0,LENGTH=19
          000023000000
          312226516060      IBFR    REAL       LOC=0,LENGTH=0
          000000000000

              $DKEND IBFR                                                        IBFR0003


NO MESSAGES FOR THIS ASSEMBLY
         SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

 CLASS  SYMBOL   VALUE   REFERENCES

        A        00003   2
        B        00016   4,13
        ERROR    00011   7
        EXIT     00015   12
        IBFR     00000
 LCTR    BLCTR
 QUAL    UNQS
 LCTR   //
        SUCCES   00013   10
```

```
   ICODE                                                          07/31/65
  7094 RELMOD ASSEMBLY.

        $IBLDR ICODE                                                                          ICOD0000
 ASSEMBLED TEXT.

        $TEXT  ICODE                                                                          ICOD0001
                                        *ICODE    MICHAEL SYNGE         6-9247     ASSEMBLED  21 AUG 64
                                        *            TO VERIFY THAT THE GIVEN WORD
                                        *            IS INDEED THE FIRST CODE OF AN
                                        *            ARRAY AND TO CONVERT ITS PREFIX
                                        *            INTO AN ADDRESS  INTEGER. THE
                                        *            RESULT IS  -1  IF THE WORD IS
                                        *            NOT THE EXPECTED CODE.
                                        *
                                        *                       CALLING SEQUENCE
                                        *
                                        *                            I=ICODE(A)
                                        *
                                              ENTRY  ICODE
                                        *

BINARY CARD (NOT PUNCHED)
       00000  0560 60 4 00003  10000   ICODE LDQ*  3,4
       00001  0754 00 0 00000  10000         PXA   0,0
       00002  4763 00 0 00003  10000         LGL   3
       00003  0600 00 0 00016  10001         STZ   TEMP
       00004  0621 00 0 00016  10001         STA   TEMP
       00005  4130 00 0 00000  10000         XCL
       00006  4320 00 0 00017  10001         ANA   =0777777000000
       00007  4340 00 0 00020  10001         LAS   =07000000
       00010  0020 00 0 01002  10011         TRA   *+2
       00011  0020 00 0 00014  10001         TRA   OK
       00012  0500 00 0 00021  10001         CLA   =-1
       00013  0020 00 0 00015  10001         TRA   EXIT
       00014  0500 00 0 00016  10001   OK    CLA   TEMP
       00015  0020 00 4 00001  10000   EXIT  TRA   1,4
                                       *
       00016  0 00000 0 00000  10000   TEMP
       00017  777777000000     10000         *LORG
       00020  000007000000     10000
       00021  400000000001     10000
                        00000  01111         END
```

```
       ICODE                                                   07/31/65
     CONTROL DICTIONARY

          $CDICT ICODE                                                   ICOD0002


BINARY CARD (NOT PUNCHED)
          000022000000           PREFACE     START=0,LENGTH=18,TYPE=7094,CMPLX=5
          000004000005
          312346242560   ICODE   DECK        LOC=0,LENGTH=18
          000022000000
          312346242560   ICODE   REAL        LOC=0,LENGTH=0
          000000000000

          $DKEND ICODE                                                   ICOD0003


NO MESSAGES FOR THIS ASSEMBLY
     SYMBOL REFERENCE DATA


REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         EXIT     00015   13
         ICODE    00000
         OK       00014   11
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
         TEMP     00016   3,4,14
```

```
       IDLETE                                                                07/31/65
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

CIDLETE     MICHAEL SYNGE          6-9247      COMPILED  21 AUG 64
C  TO DELETE AN ARRAY FROM A BUFFER                                  IDLE0030
C                                                                    IDLE0050
      FUNCTION IDLETE(ARRAY)                                         IDLE0060
C                                                                    IDLE0070
      DIMENSION ARRAY(1)                                             IDLE0080
C                                                                    IDLE0100
      M=-IBFR(ARRAY)                                                 IDLE0110   ,1
      IF (M)90,90,10                                                 IDLE0120   ,2
   10 I=32767                                                        IDLE0130   ,3
      ARRAY(I)=ARFRE(M)                                              IDLE0140   ,4
      IDLETE=0                                                       IDLE0150   ,5
      GO TO 100                                                      IDLE0160   ,6
   90 IDLETE=1                                                       IDLE0170   ,7
  100 RETURN                                                         IDLE0180
C                                                                    IDLE0190
C                                                                    IDLE0200   ,8
      END                                                            IDLE0210   ,9
```

```
        $IBLDR INIBFR                                                          INIB0000
ASSEMBLED TEXT.

        $TEXT  INIBFR                                                          INIB0001
                                  *INIBFR     MICHAEL SYNGE          6-9247       ASSEMBLED  21 SEP 64
                                  *
                                  *
                                  *  TO INITIALIZE ANY NUMBER OF STORAGE BUFFERS
                                  *
                                  *    CALLING SEQUENCE -
                                  *
                                  *                CALL INIBFR(BUFA,NBUFA,BUFB,NBUFB,.......)
                                  *
                                        ENTRY   INIBFR
                                  *

BINARY CARD (NOT PUNCHED)
       00000  0634 00 4 00004  10001  INIBFR SXA    EXIT,4
       00001  0500 00 4 00003  10000         CLA    3,4
       00002  4320 00 0 00026  10001         ANA    =0777777700000
       00003  0100 00 0 00006  10001         TZE    NEXT
       00004  0774 00 4 00000  10000   EXIT  AXT    0,4
       00005  0020 00 4 00001  10000         TRA    1,4
                                  *
                                  *
       00006  0500 00 4 00003  10000   NEXT  CLA    3,4
       00007  0400 00 0 00027  10001         ADD    =1
       00010  0621 00 0 00017  10001         STA    ARR1
       00011  0400 00 0 00027  10001   ODD   ADD    =1
       00012  0621 00 0 00024  10001         STA    ARR2
       00013  0500 60 4 00004  10000         CLA*   4,4            BUFFER LENGTH
       00014  4501 00 0 00030  10001         ORA    =0400000700000
       00015  0602 60 4 00003  10000   LENG  SLW*   3,4            FIRST CODE WORD OF BUFFER
       00016  0500 00 0 00031  10001         CLA    =0700004
       00017  0601 00 0 00000  10000   ARR1  STO    **
       00020  0500 60 4 00004  10000         CLA*   4,4
       00021  0402 00 0 00032  10001         SUB    =3
       00022  0767 00 0 00022  10000         ALS    18

BINARY CARD (NOT PUNCHED)
       00023  0600 60 0 00024  10001         STZ*   ARR2
       00024  0622 00 0 00000  10000   ARR2  STD    **             SECOND CODE WORD OF FIRST ARRAY
       00025  1 77776 4 02001  10011         TXI    INIBFR+1,4,-2
                                  *
       00026  777777700000     10000         *LORG
       00027  000000000001     10000
       00030  400000700000     10000
       00031  000000700004     10000
       00032  000000000003     10000
                      00000    01111         END
```

```
        INIBFR                                                        07/31/65
      CONTROL DICTIONARY

            $CDICT INIBFR                                                                 INIB0002

BINARY CARD (NOT PUNCHED)
        000033000000              PREFACE    START=0,LENGTH=27,TYPE=7094,CMPLX=5
        000004000005
        314531222651    INIBFR DECK          LOC=0,LENGTH=27
        000033000000
        314531222651    INIBFR REAL          LOC=0,LENGTH=0
        000000000000

            $DKEND INIBFR                                                                 INIB0003


NO MESSAGES FOR THIS ASSEMBLY
      SYMBOL REFERENCE DATA


REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         ARR1     00017   10
         ARR2     00024   12,23
         EXIT     00004   0
         INIBFR   00000   25
         LENG     00015
         NEXT     00006   3
         OCD      00011
  LCTR    BLCTR
  QUAL    UNQS
  LCTR   //
```

```
      SUBROUTINE INOUT(NTAPEO,KASE,CPCALC,POLAR,THICK,VCUT,XMACH,SEMIS,
     1XP,ZP,CBAR,RFAREA,SYM)
C     PRINTS OUT ALL INPUT AERODYNAMIC DATA

      WRITE (NTAPEO,6010)                                            1
      IF (SYM) 110,100,110
  100 WRITE (NTAPEO,6015)                                            3
      GO TO 120
  110 WRITE (NTAPEO,6020)                                            5
  120 CONTINUE

      GO TO (130,140,150),KASE
  130 WRITE (NTAPEO,6025)                                            8
      GO TO 160
  140 WRITE (NTAPEO,6030)                                           10
      GO TO 160
  150 WRITE (NTAPEO,6035)                                           12
  160 CONTINUE

      IF (CPCALC-1.) 170,180,185
  170 WRITE (NTAPEO,6040)                                           15
      GO TO 190
  180 WRITE (NTAPEO,6045)                                           17
      GO TO 190
  185 WRITE (NTAPEO,6047)                                           19
  190 CONTINUE

      IF (POLAR) 210,200,210
  200 WRITE (NTAPEO,6050)                                           22
      GO TO 220
  210 WRITE (NTAPEO,6055)                                           24
  220 CONTINUE

      IF (THICK) 240,230,240
  230 WRITE (NTAPEO,6060)                                           27
      GO TO 250
  240 WRITE (NTAPEO,6065)                                           29
  250 CONTINUE

      IF (VCUT) 270,260,270
  260 WRITE (NTAPEO,6070)                                           32
      GO TO 280
  270 WRITE (NTAPEO,6075)                                           34
  280 CONTINUE

      WRITE (NTAPEO,6095) XMACH                                     35
      WRITE (NTAPEO,6090) XP,ZP,CBAR                                36
      WRITE (NTAPEO,6085) RFAREA                                    37
      WRITE (NTAPEO,6080) SEMIS
                                                                    38
 6010 FORMAT(1H0//30H DESCRIPTION OF CASE REQUESTED)
```

```
6015 FORMAT(1H0/85H UNSYMMETRICAL CONFIGURATION - PANELS LOCATED ONLY O
    1N ONE SIDE OF X-Z PLANE(SYM = 0.))
6020 FORMAT(1H0/80H SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH
    1SIDES OF X-Z PLANE(SYM = 1.))
6025 FORMAT(1H0,9HCASE = 1.,10X,35HCALCULATE WING SHAPE, GIVEN WING CL)
6030 FORMAT(1H0,9HCASE = 2.,10X,25HCALCULATE CL, GIVEN SHAPE)
6035 FORMAT(1H0,9HCASE = 3.,10X,19HOPTIMIZE WING SHAPE)
6040 FORMAT(1H0,11HCPCALC = 0.,8X,9HLINEAR CP)
6045 FORMAT(1H0,11HCPCALC = 1.,8X,13HNON-LINEAR CP)
6047 FORMAT(1H0,11HCPCALC = 2.,8X,8HEXACT CP)
6050 FORMAT(1H0,10HPOLAR = 0. ,9X,20HPOLARS NOT REQUESTED)
6055 FORMAT(1H0,10HPOLAR = 1. ,9X,16HPOLARS REQUESTED)
6060 FORMAT(1H0,10HTHICK = 0.,9X,40HWING THICKNESS PRESSURES NOT TO BE
    1ADDED)
6065 FORMAT(1H0,10HTHICK = 1.,9X,36HWING THICKNESS PRESSURES TO BE ADDE
    1D)
6070 FORMAT(1H0,9HVOUT = 0.,10X,37HVELOCITY COMPONENTS NOT TO BE PRINTE
    1D)
6075 FORMAT(1H0,9HVOUT = 1.,10X,33HVELOCITY COMPONENTS TO BE PRINTED)
6080 FORMAT (1H0,17HWING SEMI-SPAN = ,F10.4)
6085 FORMAT(1H0,21HWING REFERENCE AREA =,F10.4)
6090 FORMAT(1H0,48HPOINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED/16H
    1  X-COORDINATE =,F10.4/16H  Z-COORDINATE =,F10.4//
    225H REFERENCE CHORD LENGTH =,F10.4)
6095 FORMAT(1H0,13HMACH NUMBER =,F10.4)

     RETURN
```

```
      SUBROUTINE INPUTB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION REC(5000),LREC(10)

C     SUBROUTINE TO INPUT DATA FOR BODY PANELING
C     SUBROUTINES
      ND4=ND2(1)

C     READ CARD INPUT - PLANB,PLANW ARE CODES FOR PLANES
C     THAT INTERSECT BODY      PLANB IS NUMBER OF CUTTING
C     PLANES TO INTERSECT FORE AND AFT OF BODY-WING
C     INTERSECTION REGION      PLANW IS NUMBER OF CUTTING
C     PLANES TO INTERSECT IN THE BODY-WING INTERSECTION
C     REGION      PLANE IS TOTAL NUMBER OF PLANES
C     TO INTERSECT BODY AND DEFINE PANEL LEADING AND
C     TRAILING EDGES      TOLB IS TOLERANCE USED
C     IN PROGRAM CHECK FOR SMALL SUBPANELS      IF SLOPE
C     OF SUBPANEL EDGE FROM STREAMWISE IS LESS THAN
C     TOLERANCE, SUBPANEL IS OMITTED FROM OUTPUT                   ,1
   10 READ (NREAD,900) PLANB,PLANW,TOLB                            ,2    ,3    ,4
      WRITE (NWRITE,950)

C     DEFINE CONSTANTS                                             ,5    ,6
      NPLNB=PLANB                                                  ,7
      IF (NPLNB .LT. 0 .OR. NPLNB .GT. 21) GO TO 298               ,8    ,9    ,10
      NPLNW=PLANW                                                  ,11
      IF (NPLNW .LT. 0 .OR. NPLNW .GT. 16) GO TO 298               ,12   ,13   ,14
      NPLANE=NPLNB+NPLNW                                           ,15
      IF (NPLANE .LT. 2 .OR. NPLANE .GT. 21) GO TO 298             ,16   ,17   ,18
      NPLN1=NPLANE-1                                               ,19
      KOPTB=0                                                      ,20
      SYM=TOLB                                                     ,21
      IF (KOPTB) 300,15,200                                        ,22
   15 IF (NPLANE) 300,300,20                                       ,23
   20 IF (NPLNB) 300,35,25

C     READ CARD INPUT - X-AXIS INTERCEPT VALUES OF
C     BODY CUTTING PLANES TO INTERSECT FORE AND AFT OF
C     BODY-WING INTERSECTION REGION                                ,24
   25 READ (NREAD,900) (XCEPTB(I),I=1,NPLNB)                       ,25   ,26   ,27   ,28   ,29
      READ (ND1) (REC(I),I=1,6)                                    ,30   ,31   ,32   ,33   ,34
      A=REC(5)                                                     ,35
      B=REC(6)
```

```
C     CHANGE X-AXIS INTERCEPT VALUES TO ALLOW FOR
C     BODY CAMBER                                              ,36
      DO 30 I=1,NPLNB                                          ,37
 30   XCEPTB(I)=A*XCEPTB(I)+B                                  ,38   ,39
      CALL CHECK (XCEPTB,NPLNB,2,L,NWRITE)                     ,40
      IF (L) 298,35,298

C     READ TAPE INPUT - BODY-WING INTERSECTION                 ,41
 35   NREC=4                                                   ,42
      KCHK=NREC+1                                              ,43
      CALL FSR (NREC,ND4,LERROR)                               ,44
      READ (ND4) (LREC(J),J=1,10)                              ,45   ,46   ,47   ,48   ,49
      KREC=LREC(1)                                             ,50
      IF (KREC-KCHK) 293,40,293                                ,51
 40   KCODE=LREC(3)

C     DEFINE KINT      IF KINT = 0, BODY-WING INTER-
C     SECTION HAS NOT BE REQUESTED OR CALCULATED
C     IF KINT = 1, OTHERWISE                                   ,52
      IF (KCODE) 293,44,43                                     ,53
 43   KINT=0                                                   ,54
      GO TO 115                                                ,55
 44   KINT=1                                                   ,56
      NWORD=LREC(2)                                            ,57
      IF (NWORD-5000) 45,45,294                                ,58
 45   NPER=LREC(4)                                             ,59
      IF (NPER-16) 46,46,294

C     READ X-, Y- AND Z-COORDINATES OF
C     BODY-WING INTERSECTION                                   ,60
 46   READ (ND4) (REC(J),J=1,NWORD)                            ,61   ,62   ,63   ,64   ,65
      DO 47 J=1,NPER                                           ,66
      XI(J)=REC(3*J-2)                                         ,67
      YI(J)=REC(3*J-1)                                         ,68
 47   ZI(J)=REC(3*J)                                           ,69   ,70
 50   IF (NPLNW) 300,115,53

C     READ CARD INPUT - CODES INDICATING WING PERCENT
C     LINES THAT ARE TO DEFINE BODY-CUTTING PLANES TO
C     INTERSECT IN BODY-WING INTERSECTION REGION

C     DEFINE CONSTANT NLOOP TO BE NUMBER OF CARDS
C     TO BE READ                                               ,71
   53 NLOOP=((NPLNW-1)/6)+1                                    ,72
      DO 57 J=1,NLOOP                                          ,73
      JSTART=1+(J-1)*6                                         ,74
      JEND=6+(J-1)*6                                           ,75
      READ (NREAD,900) (CODEBW(K),K=JSTART,JEND)               ,76   ,77   ,78   ,79   ,80
      IF (J-1) 300,55,57

C     CHECK FOR CONSTANT OF -1 IN FIRST FIELD OF
C     FIRST CARD                                               ,81
 55   IF (CODEBW(1)) 65,300,57                                 ,82
```

```
              INPUTB                                                   09/14/65
         EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

  57     CONTINUE

 C       DEFINE X-AXIS INTERCEPTS OF BODY CUTTING PLANES
 C       THAT INTERSECT IN BODY-WING INTERSECTION REGION                      ,83    ,84
  60     DO 63 J=1,NPLNW                                                      ,85
         K=CODEBW(J)+.001                                                     ,86
         XCEPTW(J)=XI(K)                                                      ,87
         YCEPTW(J)=YI(K)                                                      ,88
  63     ZCEPTW(J)=ZI(K)                                                      ,89    ,90
         GO TO 70                                                             ,91
  65     DO 67 J=1,NPLNW                                                      ,92
         XCEPTW(J)=XI(J)                                                      ,93
         YCEPTW(J)=YI(J)                                                      ,94
  67     ZCEPTW(J)=ZI(J)

 C       REDEFINE BODY-CUTTING PLANES                                         ,95    ,96
  70     IF (NPLNB) 300,105,75                                                ,97
  75     IF (NPLNW) 300,115,80                                                ,98
  80     JLEAD=0                                                              ,99
         DO 100 J=1,NPLANE                                                    ,100
         IF (XI(1)-XCEPTB(J)) 90,297,85                                       ,101
  85     XCEPT(J)=XCEPTB(J)                                                   ,102
         JLEAD=J                                                              ,103
         GO TO 100                                                            ,104
  90     DO 93 K=1,NPLNW                                                      ,105
         L=JLEAD+K                                                            ,106
  93     XCEPT(L)=XCEPTW(K)                                                   ,107   ,108
         IF (XI(NPER)-XCEPTB(JLEAD+1)) 95,297,297                             ,109
  95     JTRAIL=JLEAD+NPLNW                                                   ,110
         JLEFT=NPLANE-JTRAIL                                                  ,111
         DO 97 K=1,JLEFT                                                      ,112
         L=JTRAIL+K                                                           ,113
         M=JLEAD+K                                                            ,114
  97     XCEPT(L)=XCEPTB(M)                                                   ,115   ,116
         GO TO 130                                                            ,117
 100     CONTINUE                                                             ,118   ,119
         GO TO 130                                                            ,120
 105     IF (NPLANE-NPLNW) 300,107,300                                        ,121
 107     DO 110 J=1,NPLNW                                                     ,122
 110     XCEPT(J)=XCEPTW(J)                                                   ,123   ,124
         GO TO 130                                                            ,125
 115     IF (NPLANE-NPLNB) 300,117,300                                        ,126
 117     DO 120 J=1,NPLNB                                                     ,127
 120     XCEPT(J)=XCEPTB(J)                                                   ,128   ,129
 130     REWIND ND4

 C       WRITE TAPE OUTPUT - X-PRIME INTERCEPTS OF
 C       BODY CUTTING PLANES                                                  ,130
         WRITE (NWRITE,960) NPLANE,(XCEPT(J),J=1,NPLANE)

 C       READ TAPE INPUT - NPER IS NUMBER
 C       OF DEFINING BODY PERCENT LINES

 C       READ X-, Y- AND Z-COORDINATES OF POINTS ON
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)
C     BODY MERIDIAN LINES                                      ,131 ,132 ,133 ,134 ,135 ,136
  160 READ (ND4) (LREC(I),I=1,10)                              ,137 ,138 ,139 ,140 ,141
      KREC=LREC(1)                                             ,142
      IF (KREC-1) 290,163,290                                  ,143
  163 KCODE=LREC(3)                                            ,144
      IF (KCODE) 290,164,290                                   ,145
  164 NWORD=LREC(2)                                            ,146
      IF (NWORD-5000) 167,167,294                              ,147
  167 NPER=LREC(4)                                             ,148
      IF (NPER-16) 168,168,294                                 ,149
  168 NPER1=NPER-1

C     READ TAPE INPUT - NPTS IS NUMBER OF POINTS
C     ON BODY PERCENT LINES                                    ,150
      READ (ND4) (REC(I),I=1,NWORD)                            ,151 ,152 ,153 ,154 ,155
      K=-2                                                     ,156
      DO 170 I=1,NPER                                          ,157
      K=K+3                                                    ,158
  170 NPTS(I)=REC(K+1)                                         ,159 ,160
      DO 175 I=1,NPER                                          ,161
      IF (NPTS(I)-90) 175,175,294                              ,162
  175 CONTINUE                                                 ,163 ,164
      DO 180 I=1,NPER                                          ,165
      NSTOP=NPTS(I)                                            ,166
      DO 180 J=1,NSTOP                                         ,167
      K=K+3                                                    ,168
      X(I,J)=REC(K)                                            ,169
      Y(I,J)=REC(K+1)                                          ,170
  180 Z(I,J)=REC(K+2)                                          ,171 ,172 ,173
      REWIND ND4                                               ,174
      GO TO 300

C     READ CARD INPUT - X-, Y- AND Z-COORDINATES OF
C     BODY PANEL CORNER POINTS                                 ,175
  200 NPER=COLB+1.                                             ,176
      NPER1=NPER-1                                             ,177
      DO 210 I=1,NPER1                                         ,178
      DO 210 J=1,NPLN1                                         ,179
  210 READ (NREAD,900) XCOR(I,J),XCOR(I+1,J),XCOR(I,J+1),
     1XCOR(I+1,J+1),YCOR(I,J),YCOR(I+1,J),YCOR(I,J+1),YCOR(I+1,J+1),
     2ZCOR(I,J),ZCOR(I+1,J),ZCOR(I,J+1),ZCOR(I+1,J+1)          ,180 ,181 ,182 ,183 ,184
      WRITE (NWRITE,940)                                       ,185 ,186
      GO TO 300

C     ERROR MESSAGES                                           ,187
  290 KODEB=1                                                  ,188
      GO TO 300                                                ,189
  293 KODEI=1                                                  ,190
      GO TO 300                                                ,191
  294 WRITE (NWRITE,970)                                       ,192 ,193
      GO TO 290                                                ,194
  297 WRITE (NWRITE,990)                                       ,195 ,196
      GO TO 290                                                ,197
```

```
          INPUTB                                                    09/14/65
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

298  WRITE (NWRITE,980)                                                  ,198  ,199
     GO TO 290                                                           ,200
300  RETURN                                                              ,201
900  FORMAT (6F10.0)
940  FORMAT (1H1,9X,43HBODY PANELS HAVE BEEN DEFINED ON CARD INPUT)
950  FORMAT (1H1,9X,18HBODY PANEL ROUTINE)
960  FORMAT (1H0,9X,10HTHERE ARE ,I3,51H TRANSVERSE VERTICAL PLANES THA
    1T INTERSECT THE BODY/10X,42HTO DEFINE PANEL LEADING AND TRAILING E
    2DGES//10X,12HX-INTERCEPTS//(F20.5))
965  FORMAT (1H0,9X,43HBODY PANELS HAVE BEEN DEFINED ON CARD INPUT)
970  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X,
    151HIMPROPER BODY DEFINITION OR INTERSECTION DEFINITION/10X,
    250HDIMENSION EXCEEDED FOR NPTS, X, Y, Z OR REC ARRAYS)
980  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X,
    121HINPUT CARD DATA ERROR)
990  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTB/10X
    121HINPUT CARD DATA ERROR/10X
    1104HCUTTING PLANE THAT IS DEFINED BY X-AXIS INTERCEPT VALUE INTERS
    2ECTS BODY IN BODY-WING INTERSECTION REGION)
      END                                                                ,202
```

```
         INPUTW                                                      09/14/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE INPUTW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION REC(5000),LREC(10)

C     SUBROUTINE TO INPUT DATA FOR WING PANELING
C     SUBROUTINES

 10   ND4=ND2(1)                                                        ,1
      REWIND ND4                                                        ,2
      IF (IJ-1) 20,190,300

C     READ CARD INPUT - PLANE IS NUMBER OF PLANES
C     TO INTERSECT WING AND DEFINE WING PANEL
C     INBOARD AND OUTBOARD EDGES     OPTF IS CODE FOR CARD
C     INPUT OPTION     IF OPTF = 1, UPPER AND LOWER
C     SURFACE AIRFOIL COORDINATES ARE READ FROM
C     ADDITIONAL INPUT CARDS     IF OPTF = 0, OTHERWISE
C     SNUM IS NUMBER OF AIRFOIL SECTION COORDINATE
C     TABLES TO BE READ     TOLW IS TOLERANCE USED IN
C     PROGRAM CHECK FOR SMALL SUBPANELS     IF SLOPE
C     OF SUBPANEL EDGE FROM STREAMWISE IS LESS THAN
C     TOLERANCE, SUBPANEL IS OMITTED FROM OUTPUT
C     NOTE THAT ALL PANELS OF MAIN WING REGION
C     ARE ONE PART PANELS AND NO CHECK IS NECESSARY
C     HOWEVER, FOR PANELS ON INBOARD EDGE OF WING
C     IN BODY-WING CASE WITH NON-STREAMWISE INTER-
C     SECTION LINE AND FOR PANELS ON NON-STREAM-
C     WISE OUTBOARD EDGE SUBPANEL IS CHECKED AGAINST
C     TOLERANCE                                                         ,3
   20 READ (NREAD,900) PLANE,OPTF,SNUM,TOLW

C     DEFINE CONSTANTS                                                  ,4    ,5    ,6
      NPLANE=PLANE                                                      ,7
      IF (NPLANE .LT. 1 .OR. NPLANE .GT. 16) GO TO 295                  ,8    ,9    ,10
      NPLN1=NPLANE-1                                                    ,11
      KOPTW=0                                                           ,12
      KOPTF=OPTF                                                        ,13
      NUMS=SNUM                                                         ,14
      KACE=0                                                            ,15
      SYM=TOLW                                                          ,16
      IF (KOPTW) 300,25,250                                             ,17
 25   IF (NPLANE) 300,300,30                                            ,18
   30 NREC=2
```

```
          INPUTW                                                          09/14/65
       EXTERNAL FORMULA NUMBER   -    SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

C      READ CARD INPUT - Y-AXIS INTERCEPT VALUES FOR
C      PLANES THAT INTERSECT WING     NOTE THAT ALL BUT
C      LAST (OR OUTBOARD) CUTTING PLANE MUST BE
C      STREAMWISE AND Y-AXIS INTERCEPTS ARE SAME
C      AS Y-COORDINATES OF CUTTING PLANE AND LEADING
C      EDGE INTERSECTION     LAST CUTTING PLANE HAS, IN
C      GENERAL, AN ARBITRARY SLOPE AND IS DEFINED
C      BY CODE AND ARRAY OF VALUES     CPNT IS CODE
C      IF CPNT = 0, X- AND Y-COORDINATES OF LEADING
C      AND TRAILING EDGE INTERSECTIONS WITH CUTTING
C      PLANE ARE GIVEN     IF CPNT = 1, LEADING EDGE
C      INTERCEPTS AND SLOPE ARE GIVEN     IF CPNT =
C      2, TRAILING INTERCEPTS AND SLOPE ARE GIVEN                ,19
 70    READ (NREAD,900) (YCEPT(I),I=1,NPLN1)                     ,20   ,21   ,22   ,23   ,24
       READ (NREAD,900) CPNT,(VALUE(I),I=1,5)                    ,25   ,26   ,27   ,28   ,29   ,30

       CALL CHECK (YCEPT,NPLN1,2,L,NWRITE)                       ,31
       IF (L) 295,75,295                                         ,32
   75 KPNT=CPNT

C      CALCULATE OUTBOARD CUTTING PLANE SLOPE IF
C      NOT GIVEN AS INPUT                                        ,33
       IF (KPNT-1) 80,90,90                                      ,34
 80    IF (ABS(VALUE(4)-VALUE(2))-.001) 83,83,87                 ,35
 83    SLOPE=0.                                                  ,36
       GO TO 100                                                 ,37
 87    SLOPE=(VALUE(3)-VALUE(1))/(VALUE(4)-VALUE(2))             ,38
       GO TO 100                                                 ,39
 90    SLOPE=VALUE(5)

C      WRITE TAPE OUTPUT - INPUT CARD DATA                       ,40
 100   WRITE (NWRITE,910)                                        ,41   ,42
       WRITE (NWRITE,920) NPLANE                                 ,43   ,44   ,45
       WRITE (NWRITE,930) (YCEPT(I),I=1,NPLN1)                   ,46   ,47   ,48   ,49   ,50
       IF (KPNT-1) 115,115,120                                   ,51
 115   WRITE (NWRITE,935) VALUE(1),VALUE(2),SLOPE                ,52   ,53   ,54
       GO TO 125                                                 ,55
 120   WRITE (NWRITE,940) VALUE(3),VALUE(4),SLOPE                ,56   ,57   ,58
       GO TO 125                                                 ,59
 125   WRITE (NWRITE,950)

C      READ TAPE INPUT - NPTS IN NUMBER OF POINTS
C      ON ARBITRARY PERCENT LINE                                 ,60   ,61
 150   CALL FSR (NREC,ND4,LERROR)                                ,62
       KCHECK=NREC+1                                             ,63
       READ (ND4) (LREC(I),I=1,10)                               ,64   ,65   ,66   ,67   ,68
       KREC=LREC(1)                                              ,69
       IF (KREC-KCHECK) 280,160,280                              ,70
 160   KCODE=LREC(3)                                             ,71
       IF (KCODE) 280,163,280                                    ,72
 163   NWORD=LREC(2)                                             ,73
       IF (NWORD-5000) 165,165,287                               ,74
 165   NPER=LREC(4)                                              ,75
```

```
      EXTERNAL FORMULA NUMBER  -  SOURCE STATEMENT  -  INTERNAL FORMULA NUMBER(S)

      IF (NPER-16) 167,167,287                                  ,76
 167  NPER1=NPER-1

C     READ X-, Y- AND Z-COORDINATES OF POINTS ON
C     WING PERCENT CHORD LINES                                  ,77
      READ (ND4) (REC(I),I=1,NWORD)                             ,78   ,79   ,80   ,81   ,82
      K=-2                                                      ,83
 175  DO 180 J=1,NPER                                           ,84
      K=K+3                                                     ,85
 180  NPTS(J)=REC(K+1)                                          ,86   ,87
      DO 183 J=1,NPER                                           ,88
      IF (NPTS(J)-90) 183,183,287                               ,89
 183  CONTINUE                                                  ,90   ,91
      DO 185 J=1,NPER                                           ,92
      NSTOP=NPTS(J)                                             ,93
      DO 185 I=1,NSTOP                                          ,94
      K=K+3                                                     ,95
      X(I,J)=REC(K)                                             ,96
      Y(I,J)=REC(K+1)                                           ,97
 185  Z(I,J)=REC(K+2)                                           ,98   ,99   ,100
      REWIND ND4                                                ,101
      GO TO 300

C     READ CARD INPUT - UPPER AND LOWER SURFACE
C     AIRFOIL COORDINATES                                       ,102
 190  IF (KOPTF) 300,290,195                                    ,103
 195  KACE=1                                                    ,104
      IF (NUMS-1) 300,200,205                                   ,105
 200  ISTOP=1                                                   ,106
      GO TO 210                                                 ,107
 205  ISTOP=NPLN1+KSTART-1                                      ,108
 210  DO 220 I=1,ISTOP                                          ,109
      DO 220 N=1,2                                              ,110
      READ (NREAD,900) XNUM(I,N)                                ,111  ,112  ,113
      NUM=XNUM(I,N)                                             ,114
 220  READ (NREAD,900) (XFOIL(I,J,N),ZFOIL(I,J,N),J=1,NUM)

C     CONVERT AIRFOIL(S) TO UNIT AIRFOIL(S)                     ,115  ,116  ,117  ,118  ,119  ,120
                                                                ,121
 225  DO 240 I=1,ISTOP                                          ,122
      DO 240 N=1,2                                              ,123
      NUM=XNUM(I,N)                                             ,124
      IF (XFOIL(I,NUM,N)-1.) 230,240,230                        ,125
 230  DO 235 J=1,NUM                                            ,126
      ZFOIL(I,J,N)=ZFOIL(I,J,N)/XFOIL(I,NUM,N)                  ,127
 235  XFOIL(I,J,N)=XFOIL(I,J,N)/XFOIL(I,NUM,N)                  ,128  ,129
 240  CONTINUE                                                  ,130  ,131  ,132
      GO TO 300

C     READ CARD INPUT - X-, Y- AND Z-COORDINATES OF
C     WING PANEL CORNER POINTS                                  ,133
 250  NPER=ROW+1.                                               ,134
      NPER1=NPER-1                                              ,135
      NPLANE=NPLANE+1                                           ,136
```

```
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      NPLN1=NPLANE-1                                                       ,137
      DO 260 I=1,NPLN1                                                     ,138
      DO 260 J=1,NPER1                                                     ,139
 260  READ (NREAD,900) XCOR(I,J),XCOR(I+1,J),XCOR(I,J+1),
     1XCOR(I+1,J+1),YCOR(I,J),YCOR(I+1,J),YCOR(I,J+1),YCOR(I+1,J+1),
     2ZCOR(I,J),ZCOR(I+1,J),ZCOR(I,J+1),ZCOR(I+1,J+1)                      ,140  ,141  ,142  ,143  ,144
      GO TO 300
C     ERROR MESSAGES                                                       ,145
 280  IJ1=IJ+1                                                             ,146
      GO TO (283,285),IJ1                                                  ,147
 283  KODEW=1                                                              ,148
      GO TO 300                                                            ,149
 285  KODEWU=1                                                             ,150
      GO TO 300                                                            ,151
 287  WRITE (NWRITE,980)                                                   ,152  ,153
      GO TO 283                                                            ,154
 290  WRITE (NWRITE,970)                                                   ,155  ,156
      GO TO 283                                                            ,157
 295  WRITE (NWRITE,990)                                                   ,158  ,159
      GO TO 283                                                            ,160
 300  RETURN                                                               ,161
 900  FORMAT (6F10.0)
 910  FORMAT (1H1,15X,18HWING PANEL ROUTINE)
 920  FORMAT (1H0,9X,10HTHERE ARE ,I3,67H VERTICAL PLANES THAT INTERSECT
     1 THE WING TO DEFINE PANEL SIDE EDGES)
 930  FORMAT (1H0,9X,12HLEADING-EDGE/10X,
     111HY-INTERCEPT//(F20.5))
 935  FORMAT (1H0,19X,22HOUTBOARD CUTTING PLANE/17X,
     112HLEADING-EDGE/10X,11HX-INTERCEPT,5X,11HY-INTERCEPT,8X,
     25HSLOPE/(F19.5,F16.5,F18.5))
 940  FORMAT (1H0,19X,22HOUTBOARD CUTTING PLANE/17X,
     113HTRAILING-EDGE/10X,11HX-INTERCEPT,5X,11HY-INTERCEPT,8X,
     25HSLOPE/(F19.5,F16.5,F18.5))
 950  FORMAT (1H0,9X,45HZERO-SLOPE INDICATES STREAMWISE CUTTING PLANE)
 960  FORMAT (1H0,9X,43HWING PANELS HAVE BEEN DEFINED ON CARD INPUT)
 970  FORMAT (1H1,9X,63HUPPER AND LOWER SURFACE AIRFOIL COORDINATES NOT
     1READ FROM CARDS)
 980  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTW/10X,
     151HIMPROPER WING DEFINITION OR INTERSECTION DEFINITION/10X,
     250HDIMENSION EXCEEDED FOR NPTS, X, Y, Z OR REC ARRAYS)
 990  FORMAT (1H1,9X,33HERROR MESSAGE - SUBROUTINE INPUTW/10X,
     121HINPUT CARD DATA ERROR)
      END                                                                  ,162
```

```
      SUBROUTINE INTAPE
C     READS IN GEOMETRICAL DATA FROM PANELING AND DEFINITION LINKS
      CCMMCN DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEC,NBODY,NWING,XMACH,SYM,KACE
      COMMON /BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     C,CHORD(210)
     I,ISYM
     N,NPART(210),NPANEL,NROW(2)
     T,THETA(210),TAIL
     U,U(210)
     V,V(210),VPM(210),VV(210),VPMM(210)
     W,W(210),WPM(210),WW(210),WPMM(210)
     X,X(210,3,4),XBAR(210),XC(210)
     Y,Y(210,3,4),YBAR(210),YC(210)
     Z,Z(210,3,4),ZBAR(210),ZC(210)
      DIMENSION JSAVE(2),NSAVE(2),XB(51),R(51),ZDELTA(51)
      DIMENSION ARRAY(3,10),THETAB(10)
      DIMENSION XYZ(3)
C     READ IN MACH NUMBER, CONFIGURATION SYMMETRY CONDITION
C     BOUNDARY CONDITION TYPE AND TAIL OPTION CODE
      READ (NTAPEI,5000) XMACH,SYM
      BCOND=0
      KCND=BCOND
      IF (SYM) 50,45,50
 45   ISYM=1
      GC TO 55
 50   ISYM=2
 55   CONTINUE
C     CODE TO INDICATE IF THE CASE INCLUDES A THICK WING
      THKW=KACE
C     TEST FOR TYPE OF CONFIGURATION
      IF (NBODY) 120,110,12C
C     WING ONLY CASE
 110  KACE=1
      JSAVE(1)=0
      NSAVE(1)=NWING
      NPANEL=NWING
      NBODYS=0
      KRG=1
      S=1.
      T=0.
      GC TO 160
  120 CONTINUE
```

```
                 INTAPE      -  EFN    SOURCE STATEMENT  -  IFN(S)  -          05/22/67

C       READ NUMBER OF BODY SOURCE SEGMENTS AND THE INDEX OF THE X -
C       COORDINATE AT LEADING EDGE OF THE WING
        READ (NTAPEB) (XYZ(I),I=1,3),ZA,S,T                                          15
        ALPHAA=XYZ(3)/XYZ(1)
        READ (NTAPEB) NBODYS,NXLE

C       READ BODY X-COORDINATES, BODY RADIUS, AND INCREMENT OF BODY CAMBER           22
        READ (NTAPEB) (XB(I),R(I),ZDELTA(I),I=1,NBODYS)                              25
        REWIND NTAPEB
                                                                                     34
        IF (NBODY) 130,999,140

C       BODY ONLY CASE
 130    KACE=2
        NPANEL=0
        NWING=0
        WRITE (NTAPEC) NBODY,NWING,XMACH,SYM,KACE                                    39
        GO TO 325

C       WING AND BODY CASE
 140    KACE=3
        JSAVE(1)=0
        NSAVE(1)=NBODY
        JSAVE(2)=NBODY
        NSAVE(2)=NWING
        NPANEL=NWING+NBODY
        NRG=2
 160    CONTINUE


        DC 210 L=1,NRG
        J=JSAVE(L)
        N=NSAVE(L)

        DC 190I=1,N
        J=J+1

C       READS IN PANEL COORDINATES
        READ (NTAPEC) NPN,NPART(J),(X(J,1,K),Y(J,1,K),Z(J,1,K),K=1,4)                54
        IF (NPART(J)-1)170,190,170
 170    NPT=NPART(J)
        DO 180M=2,NPT
        READ (NTAPEC)(X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4)                              69
 180    CONTINUE
 190    CONTINUE

        J=JSAVE(L)
        DO 200I=1,N
        J=J+1

C       READS IN COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C       ALPHAS, AND CHORD LENGTHS
        READ (NTAPEC) NPN,XBAR(J),YBAR(J),ZBAR(J),XC(J),YC(J),ZC(J),AREA(J
```

```
     1),THETA(J),ALPHAS(J),CHORD(J)                                          84
 200  CONTINUE

C     READS IN THE NUMBER OF PANELS IN A COLUMN AND THE LOCATION OF THE
C     CCNTROL POINT
      READ (NTAPEC) NROW(L),XCTP                                             97
  210 CCNTINUE
      REWIND NTAPEC
                                                                             101
  240 CCNTINUE
      DO 250 J=1,NWING
      JJ=J+NBODY

C     STORES THICKNESS SLOPES AND CAMBER SLOPES
      ALPHAT(J)=THETA(JJ)
 250  ALPHAC(J)=ALPHAS(JJ)
      NS=NBCDY+1
      DO 300 J=NS,NPANEL
      THETA(J)=0.
      ALPHAS(J)=0.
 300  CONTINUE
      WRITE (NTAPEC) NBODY,NWING,XMACH,SYM,KACE


C     WRITES COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C     ALPHAS, CHORD LENGTHS, NUMBER OF REGIONS, NUMBER OF ROWS PER
C     CCLUMN IN A REGION, AND CONTROL POINT LOCATION                         122
      WRITE (NTAPEC) (I,XBAR(I),YBAR(I),ZBAR(I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),ALPHAS(I),CHORD(I),I=1,NPANEL),NRG,(NROW(I),I=1,NRG),XC
     2TP                                                                     123
      END FILE NTAPEC

C     WRITE THICKNESS SLOPES                                                 145
      WRITE (NTAPEC) (ALPHAT(I),I=1,NWING)                                   146
      END FILE NTAPEC

C     WRITE CAMBER SLOPES                                                    153
      WRITE (NTAPEC) (ALPHAC(I),I=1,NWING)                                   154
      END FILE NTAPEC
                                                                             161
      IF (NBODY) 325,510,325
 325  CONTINUE

C     WRITE SLOPE OF BODY AXIS WITH RESPECT TO THE DEFINING AXIS
      WRITE (NTAPEC) ALPHAA

C     WRITE NUMBER OF BODY SOURCE SEGMENTS, INDEX OF X- COORDINATE AT
C     LEADING EDGE OF WING, HEIGHT OF WING PLANE ABCVE BODY AXIS, X
C     CCORDINATE OF BOCY                                                     163
      WRITE (NTAPEC) NBODYS,NXLE,ZA,(XB(I),I=1,NBODYS)                       164
      IF (NWING) 400,350,400
  350 CCNTINUE

C     FCR BODY ONLY CASE, READ NUMBER OF THETAS REPRESENTING BODY
```

```
            INTAPE      -  EFN    SOURCE STATEMENT  -  IFN(S)  -

      READ (NTAPED) NDUMMY,NDUMMY,NDUMMY,NTHETA

C     READ BODY THETAS                                                   172
      READ (NTAPED) ((ARRAY(I,J),I=1,3),J=1,NTHETA)                      177
      REWINC NTAPED                                                      187
      DO 375 J=1,NTHETA
  375 THETAB(J)=ARRAY(1,J)
      GC TC 450

C     FOR WING-BODY CASE, COMPUTE NUMBER OF THETAS AND THETAS FROM
C     PANEL REPRESENTATION OF BODY
  400 CONTINUE
      NROWB=NROW(1)
      NTHETA=NBODY/NROWB
      DO 425 J=1,NTHETA
      JJ=(J-1)*NROWB+1
  425 THETAB(J)=ABS(THETA(JJ))*57.2957795
  450 CONTINUE

C     WRITE NUMBER OF THETAS, AND THETAS
      WRITE (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)

C     WRITE RADIUS AND CAMBER INCREMENT OF BODY                          208
      WRITE (NTAPEC) (R(I),I=1,NBODYS)                                   215
      WRITE (NTAPEC) (ZDELTA(I),I=1,NBODYS)                              222
      WRITE (NTAPEC) (ZDELTA(I),I=1,NBODYS)                              229
      END FILE NTAPEC
                                                                         236
      IF (NBODY) 500,500,550
  500 CONTINUE
      END FILE NTAPEC                                                    238
      END FILE NTAPEC                                                    239
  510 CONTINUE
      END FILE NTAPEC

C     STORE GEOMETRY DATA ON SCRATCH/SAVE TAPE
C     FCR USE IN FLOW VISUALIZATION LINK                                 240
  550 CONTINUE
      NWINGS=NWING+NWING/NROW(NRG)
      WRITE (NTAPEC) KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS
     $,NROW,XMACH,SYM                                                    243
      END FILE NTAPEC
                                                                         245
      IF (NPANEL) 600,600,575
  575 CONTINUE
      WRITE (NTAPEC) NPANEL                                              247
      DC 580 J=1,NPANEL
      WRITE (NTAPEC) NPART(J)                                            250
      NPT=NPART(J)
      WRITE (NTAPEC) ((X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4),M=1,NPT)       253
  580 CONTINUE
      WRITE (NTAPEC) (XBAR(J),YBAR(J),ZBAR(J),J=1,NPANEL)                265
      WRITE (NTAPEC) (XC(J),YC(J),ZC(J),J=1,NPANEL)                      274
      WRITE (NTAPEC) (ALPHAS(J),THETA(J),CHORD(J),J=1,NPANEL)            283
```

```
  600 CCNTINUE
      END FILE NTAPEC

 999  CONTINUE
 5000 FCRMAT(7F10.0)
 5200 FCRMAT(I3)
 5100 FORMAT(6F10.0)
      RETURN
      END
```

```
       INTPCL                                                        08/16/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE INTPCL (NM,NROW,RATIOX,RATIOC,CHORD,ALPHAM,ALPHAD)

C     INTERPOLATES FOR NORMAL VELOCITY COMPONENT AT A SPECIFIED PERCENT
C     OF PANEL CHORD

      DIMENSION CHORD(1),ALPHAM(1),ALPHAD(1)

      NCOL=NM/NROW                                                   ,1
      R=RATIOC-RATIOX                                                ,2
      RR=-R                                                          ,3
      J=0
                                                                     ,4
      DO 100 K=1,NCOL                                                ,5
      J=J+1                                                          ,6
      ALPHAD(J)=ALPHAM(J)+R*(ALPHAM(J+1)-ALPHAM(J))/(1.+RATIOX
     1*(CHORD(J+1)/CHORD(J)-1.))                                     ,7
      DO 100 I=2,NROW                                                ,8
      J=J+1                                                          ,9
      CR=CHORD(J)/CHORD(J-1)                                         ,10
      ALPHAD(J)=ALPHAM(J)+CR*R*(ALPHAM(J)-ALPHAM(J-1))/(1.+
     1RATIOX*(CR-1.))                                                ,11
  100 CONTINUE
                                                                     ,12   ,13   ,14
      RETURN                                                         ,15
      END                                                            ,16
```

```
        INTURP                                                                07/31/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

        FUNCTION INTURP(IDIC,NDIC,LI,LO)

C       READ AND WRITE A COMMAND CARD,SEE IF COMMAND (IN THE
C       FIRST WORD) IS IN THE DICTIONARY.

        DIMENSION IDIC(1),CCM(14)
        EQUIVALENCE (COM,ICOM)

C       IDIC (INSTEAD OF DIC) IS USED AS AN ARGUMENT
C       SO THAT A FIXED-POINT COMPARISON CAN BE MADE

        READ (LI,110)COM                                       ,1    ,2    ,3
  110   FORMAT(13A6,A2)
        IF(LO)140,140,115                                      ,4
  115   WRITE (LO,120)COM                                      ,5    ,6    ,7
  120   FORMAT(/1H014A6)
  140   DO 200 INTERP=1,NDIC                                   ,8
        IF (IDIC(INTERP)-ICOM)200,500,200                      ,9
  200   CONTINUE
                                                               ,10   ,11
        INTERP=0                                               ,12
  500   INTURP=INTERP                                          ,13
        RETURN                                                 ,14
        END                                                    ,15
```

```
ASSEMBLED TEXT.

        $TEXT  INV4S                                                               INV40001

                               *INV4S   SINGLE PRECISION 7094 INVERSION ROUTINE (ADAPTION OF INV4D)
                               *              CALLING SEQUENCE -
                               *              CALL INV4S,A,N,ERR1,ERR2,SCALE,DET,DETXP
                               *              A = STARTING ADDRESS OF MATRIX
                               *                  N = NUMBER OF ROWS (COLUMNS) OF MATRIX
                               *                ERR1 = 0 IF INVERSION SUCCESSFUL
                               *                     = 1 IF OVERFLOW OCCURS
                               *                     = 2 IF MATRIX IS SINGULAR
                               *                     = 3 IF SCALED INVERSE CANNOT BE RE-SCALED
                               *                     = 4 IF ROWS AND COLUMNS CANNOT BE RE-ARRANGED
                               *                         THIS IS BASICALLY A MACHINE ERROR
                               *                     = 10 + I IF ERROR CODE 3 OCCURED SUBSEQUENT
                               *                              TO ERROR CODE I
                               *                     = 20 + I IF ERROR CODE 4 OCCURED SUBSEQUENT
                               *                              TO ERROR CODE I
                               *                ERR2 = CURRENT REDUCTION STAGE IF ERR1=1
                               *                     = RANK OF MATRIX IF ERR1 =2
                               *                     = 0 OTHERWISE
                               *               SCALE = 0 IF ERR1 NOT = 3
                               *                     = SCALING FACTOR IF ERR1=3
                               *                 DET= DECIMAL PART OF DETERMINANT IF ERR1.
                               *                      NOT = 2 - A NUMBER GT OR E TO 1. AND LT 10.
                               *                     = 0 IF ERR1=2
                               *               DETXP = EXPONENT PART OF DETERMINANT IF ERR1
                               *                      NOT = 2 - AN ADDRESS INTEGER GIVING
                               *                      POWER OF TEN FOR DETERMINANT
                               *                      DETERMINANT = DET*10.**DETXP
                                        ENTRY   INV4S
                               *
                               *              SET UP CONSTANTS AND INITIALIZE CELLS
                               *

BINARY CARD ID. INV40002
        00000  0760 00 0 00016  10000  INV4S LMTM
        00001  4760 00 0 00002  10000        EFTM
        00002  0634 00 1 00361  10001        SXA    XR1,1
        00003  0634 00 2 00362  10001        SXA    XR2,2
        00004  0634 00 3 00363  10001        SXA    XR3,3
        00005  0634 00 4 00364  10001        SXA    XR4,4
        00006  0634 00 5 00365  10001        SXA    XR5,5
        00007  0634 00 6 00366  10001        SXA    XR6,6
        00010  0634 00 7 00367  10001        SXA    XR7,7
        00011  0600 00 0 00000  10000        STZ    0              CLEAR CELL 0 FOR FPT
        00012  0600 00 0 00466  10001        STZ    CHRCNG         CLEAR CELL FOR CHARACTERISTIC SCALING
        00013  0600 00 0 00467  10001        STZ    ERRCD1         CLEAR ERROR CODE CELLS
        00014  0600 00 0 00470  10001        STZ    ERRCD2
        00015  4500 00 0 00010  10000        CAL    8              SAVE CELL 8 FOR (FPT)
        00016  0602 00 0 00443  10001        SLW    TEMP8
        00017  4500 00 0 00444  10001        CAL    FPTTRA         STORE TRANSFER FOR FPT ANALYSIS.
        00020  0602 00 0 00010  10000        SLW    8
        00021  0600 60 4 00007  10000        STZ*   7,4            ZERO SCALING CELL
        00022  0600 00 0 00451  10001        STZ    DETXP          ZERO DETERMINANT EXPONENT
```

```
         INV4S                                                      08/17/65
      ASSEMBLED TEXT.

BINARY CARD ID. INV40003
         00023  0560 60 4 00004  10000        LDQ*     4,4
         00024  4600 00 0 00452  10001        STQ      N
         00025  0200 00 0 00452  10001        MPY      N
         00026  0131 00 0 00000  10000        XCA
         00027  0601 00 0 00453  10001        STO      NSQ          N**2
         00030  0361 00 4 00003  10000        ACL      3,4
         00031  0621 00 0 00461  10001        STA      APLNSQ        A + N**2
         00032  0361 00 0 00452  10001        ACL      N
         00033  0621 00 0 00462  10001        STA      ADRINT       ADDRESS OF ROW INTEGERS +N
         00034  0361 00 0 00452  10001        ACL      N
         00035  0621 00 0 00463  10001        STA      ADCINT       ADDRESS OF COLUMN INTEGERS +N
         00036  0621 00 0 00447  10001        STA      MULTPL       ADDRESS OF REDUCTION FACTOR
         00037  0400 00 0 00733  10001        ADD      =1
         00040  0621 00 0 00446  10001        STA      DET          ADDRESS OF DETERMINANT
         00041  0400 00 0 00733  10001        ADD      =1
         00042  0621 00 0 00450  10001        STA      TEN          ADDRESS OF FLOATING POINT 10
         00043  0500 00 4 00003  10000        CLA      3,4
         00044  0621 00 0 00454  10001        STA      ADA          ADDRESS OF A
         00045  0621 00 0 00455  10001        STA      ADPVEL       ADDRESS OF CURRENT PIVOT ELEMENT

BINARY CARD ID. INV40004
         00046  0361 00 0 00452  10001        ACL      N
         00047  0621 00 0 00456  10001        STA      ADPVRW       ADDRESS OF CURRENT PIVOT ROW +   N
         00050  0534 00 1 00452  10001        LXA      N,1
         00051  4634 00 1 00652  10001        SXD      REDUC4,1     STUFF LOOP CONTROL FOR REDUCE
         00052  0500 00 0 00734  10001        CLA      =1.
         00053  0601 60 0 00446  10001        STO*     DET          START DET OFF AT 1.
         00054  0500 00 0 00735  10001        CLA      =10.
         00055  0601 60 0 00450  10001        STO*     TEN          LOAD 10 IN DOUBLE PRECISION CELL
         00056  0534 00 7 00452  10001        LXA      N,7
         00057  4634 00 7 00204  10001        SXD      RDLPED,7     TEST AT END OF REDUCTION LOOP
         00060  4634 00 7 00067  10001        SXD      STOINT,7     PUT INTEGERS INTO ROW AND COLUMN
         00061  0754 00 0 00000  10000        PXA      0,0
         00062  0774 00 1 00001  10000        AXT      1,1          INTERCHANGE ARRAYS
         00063  0754 00 1 00000  10000        PXA      0,1
         00064  0601 60 0 00463  10001        STO*     ADCINT       ADCINT HAS A TAG OF 1
         00065  0601 60 0 00462  10001        STO*     ADRINT       ADRINT HAS A TAG OF 1
         00066  1 00001 1 01001  10011        TXI      *+1,1,1      INTEGERS ARE STORED  BACKWARDS
         00067  7 00000 1 41004  10011 STOINT TXL      *-4,1,**     N IN DECREMENT
                                       *
                                       *          START OF REDUCTION LOOP
                                       *          REDUCTION STAGE IS KEPT IN XR1
                                       *          AT THE END OF THIS LOOP ALL INVERSION
                                       *             ARITHMETIC IS DONE
                                       *
         00070  0774 00 1 00001  10000        AXT      1,1          INITIALIZE REDUCTION STAGE

BINARY CARD ID. INV40005
         00071  0074 00 4 00546  10001 RDLPST TSX      FINDLE,4     FIND LARGEST ELEMENT IN N-XR1+1 SUB MATRIX
         00072  0634 00 1 00473  10001        SXA      TEMPXR,1
         00073  0754 00 1 00000  10000        PXA      0,1
         00074  0340 00 0 00464  10001        CAS      LGSTRW       DO WE INTERCHANGE ROWS
         00075  0020 00 0 01002  10011        TRA      *+2          YES
```

```
       00076  0020 00 0 00115  10001         TRA     EXCHCL        NO                                       102
       00077  0500 60 0 00446  10001         CLA*    DET           YES,ADJUST DET. SIGN                     103
       00100  0760 00 0 00002  10000         CHS                                                            104
       00101  0601 60 0 00446  10001         STO*    DET                                                    105
       00102  0074 00 4 00476  10001         TSX     INRCHR,4      EXCHANGE ROWS TO GET LGST NO ON PIV POS. 106
105    00103  0074 00 0 00473  10001         TSX     TEMPXR                                                 107
106    00104  0074 00 0 00464  10001         TSX     LGSTRW                                                 108
                                       *                           EXCHANGE ROW INTEGERS                    109
       00105  0500 00 0 00462  10001         CLA     ADRINT                                                 110
       00106  0402 00 0 00464  10001         SUB     LGSTRW                                                 111
       00107  0621 00 0 01002  10011         STA     *+2                                                    112
       00110  0621 00 0 01003  10011         STA     *+3                                                    113
       00111  0500 00 0 00000  10000         CLA     **            LGSTRW INTEGER                           114
       00112  0560 60 0 00462  10001         LDQ*    ADRINT        ADRINT HAS TAG OF 1                      115
       00113  4600 00 0 00000  10000         STQ     **            LGSTRW INTEGER                           116

BINARY CARD ID. INV40006
       00114  0601 60 0 00462  10001         STO*    ADRINT                                                 117
       00115  0754 00 1 00000  10000  EXCHCL PXA     0,1                                                    118
       00116  0340 00 0 00465  10001         CAS     LGSTCL        DO WE INTERCHANGE COLS.                  119
       00117  0020 00 0 01002  10011         TRA     *+2           YES                                      120
       00120  0020 00 0 00137  10001         TRA     STRTSC        NO                                       121
       00121  0500 60 0 00446  10001         CLA*    DET           YES, ADJUST DET. SIGN                    122
       00122  0760 00 0 00002  10000         CHS                                                            123
       00123  0601 60 0 00446  10001         STO*    DET                                                    124
       00124  0074 00 4 00521  10001         TSX     INRCHC,4      INTERCHANGE COLUMNS FOR LGST NO.         125
124    00125  0074 00 0 00473  10001         TSX     TEMPXR                                                 126
125    00126  0074 00 0 00465  10001         TSX     LGSTCL                                                 127
                                       *                           INTERCHANGE COLUMN INTEGERS              128
       00127  0500 00 0 00463  10001         CLA     ADCINT                                                 129
       00130  0402 00 0 00465  10001         SUB     LGSTCL                                                 130
       00131  0621 00 0 01002  10011         STA     *+2                                                    131
       00132  0621 00 0 01003  10011         STA     *+3                                                    132
       00133  0500 00 0 00000  10000         CLA     **                                                     133
       00134  0560 60 0 00463  10001         LDQ*    ADCINT        ADCINT HAS A TAG OF 1                    134
       00135  4600 00 0 00000  10000         STQ     **                                                     135
       00136  0601 60 0 00463  10001         STO*    ADCINT                                                 136
                                       *                                                                    137
                                       *             START OF CHARACTERISTIC SCALING ROUTINE                138
                                       *             THE LARGEST MAGNITUDE IN THE MATRIX WILL               139
                                       *             BE FORCED TO HAVE A CHARACTERISTIC OF 177              140
                                       *             AND ALL OTHERS WILL BE ADJUSTED ACCORDINGLY            141
                                       *                                                                    142

BINARY CARD ID. INV40007
       00137  3 00001 1 00162  10001  STRTSC TXH     SCLOOP+1,1,1                                           143
       00140  4500 60 0 00455  10001         CAL*    ADPVEL        LARGEST ELEMENT = PIVOT ELEMENT          144
       00141  4320 00 0 00736  10001         ANA     =O377000000000  MASK OUT ALL BUT CHARACTERISTIC        145
       00142  0402 00 0 00737  10001         SUB     =O177000000000  FIND DIFFERENCE FROM 177               146
       00143  0760 00 0 00002  10000         CHS                                                            147
       00144  0601 00 0 00466  10001         STO     CHRCNG        SAVE CHARACTERISTIC SCALING              148
       00145  0100 00 0 00162  10001         TZE     SCLOOP+1      BY-PASS SCALING IF NOT NEEDED            149
       00146  0534 00 7 00453  10001         LXA     NSO,7                                                  150
       00147  4520 60 0 00461  10001  FSTSCL NZT*    APLNSO        APLNSO HAS A TAG OF 7                    151
```

```
       INV4S                                                              08/17/65
     ASSEMBLED TEXT.

          00150   0020 00 0 00161   10001          TRA    SCLOOP         NUMBER IS ZERO, DO NOT SCALE
          00151   0500 00 0 00466   10001          CLA    CHRCNG         SCALE NUMBER
          00152   0401 60 0 00461   10001          ADM*   APLNSQ
          00153   0120 00 0 01003   10011          TPL    *+3
          00154   0600 60 0 00461   10001          STZ*   APLNSQ         IF NEGATIVE, UNDERFLOW OCCURRED
          00155   0020 00 0 00161   10001          TRA    SCLOOP
          00156   0560 60 0 00461   10001          LDQ*   APLNSQ         SCALING SUCCESSFUL, RESTORE SIGN
          00157   0763 00 0 00000   10000          LLS    0
          00160   0601 60 0 00461   10001          STO*   APLNSQ         STORE IN ORIGINAL LOCATION
          00161   2 00001 7 00147   10001   SCLOOP TIX    FSTSCL,7,1     END OF SCALING LOOP

  BINARY CARD ID. INV40008
          00162   0074 00 4 00612   10001          TSX    REDUCE,4       PERFORM CURRENT REDUCTION STAGE
          00163   0500 60 0 00455   10001          CLA*   ADPVEL
          00164   0074 00 4 00654   10001          TSX    SCALE,4        SCALE PIVOT ELEMENT
          00165   0131 00 0 00000   10000          XCA
          00166   0260 60 0 00446   10001          FMP*   DET
          00167   0074 00 4 00654   10001          TSX    SCALE,4        SCALE PRODUCT
          00170   0601 60 0 00446   10001          STO*   DET            RUNNING PRODUCT OF PIVOT ELEMENTS
          00171   0534 00 7 00452   10001          LXA    N,7
          00172   0500 60 0 00455   10001          CLA*   ADPVEL
          00173   0601 60 0 00447   10001          STO*   MULTPL         KEEP PIVOT ELEMENT TEMP. IN MULTPL
          00174   0500 60 0 00456   10001   ADJUST CLA*   ADPVRW         ADPVRW HAS TAG OF 7
          00175   0241 60 0 00447   10001          FDP*   MULTPL
          00176   4600 60 0 00456   10001          STQ*   ADPVRW         DIVIDE PIVOT ROW BY PIVOT ELEMENT
          00177   2 00001 7 00174   10001          TIX    ADJUST,7,1
          00200   0500 00 0 00734   10001          CLA    =1.
          00201   0241 60 0 00447   10001          FDP*   MULTPL
          00202   4600 60 0 00455   10001          STQ*   ADPVEL         PUT 1/(PIV. ELEM) INTO PIVOT POSITION
          00203   1 00001 1 01001   10011          TXI    *+1,1,1        INCREMENT TO NEW REDUCTION STAGE
          00204   3 00000 1 00215   10001   RDLPED TXH    REDOVR,1,**    N IN DECREMENT-LOOP EXIT

  BINARY CARD ID. INV40009
          00205   0500 00 0 00455   10001          CLA    ADPVEL
          00206   0400 00 0 00733   10001          ADD    =1
          00207   0400 00 0 00452   10001          ADD    N
          00210   0621 00 0 00455   10001          STA    ADPVEL         INCREMENT BY   N+1
          00211   0500 00 0 00456   10001          CLA    ADPVRW
          00212   0400 00 0 00452   10001          ADD    N
          00213   0621 00 0 00456   10001          STA    ADPVRW         INCREMENT BY   N
          00214   0020 00 0 00071   10001          TRA    RDLPST         ANOTHER REDUCTION STAGE
                                            *
                                            *             ALL ARITHMETIC IS OVER NOW REARANGE ROWS
                                            *             AND COLUMNS
                                            *
                                            *             FIRST, INTERCHANGE COLUMNS ACCORDING TO ROW TABLE
                                            *
          00215   0534 00 7 00452   10001   REDOVR LXA    N,7
          00216   0600 00 0 00471   10001          STZ    TEMP1          CLEAR FOR XR USE
          00217   0754 00 7 00000   10000          PXA    0,7
          00220   0734 00 1 00000   10000          PAX    0,1
          00221   0754 00 7 00000   10000   RWSRCH PXA    0,7
          00222   0402 60 0 00462   10001          SUB*   ADRINT         ADRINT HAS A TAG OF 1
          00223   0100 00 0 00226   10001          TZE    FOUNDR
```

```
       00224  2 00001 1 00221  10001         TIX    RWSRCH,1,1                              202
       00225  0020 00 0 00431  10001         TRA    ERR4         CANNOT FIND ROW            203
       00226  0754 00 7 00000  10000  FOUNDR PXA    0,7                                     204
       00227  0634 00 1 00471  10001         SXA    TEMP1,1                                 205

BINARY CARD ID. INV40010
       00230  0402 00 0 00471  10001         SUB    TEMP1                                   206
       00231  0100 00 0 00246  10001         TZE    RWLPED       COLUMN IS IN CORRECT PLACE 207
       00232  0634 00 7 00473  10001         SXA    TEMPXR,7                                208
       00233  0074 00 4 00521  10001         TSX    INRCHC,4                                209
208    00234  0074 00 0 00473  10001         TSX    TEMPXR       ROW INTEGER FOUND          210
209    00235  0074 00 0 00471  10001         TSX    TEMP1        LOCATION OF ROW INTEGER    211
                                      *             INTERCHANGE ROW INTEGERS.               212
       00236  0534 00 7 00473  10001         LXA    TEMPXR,7                                213
       00237  0500 00 0 00462  10001         CLA    ADRINT                                  214
       00240  0402 00 0 00473  10001         SUB    TEMPXR                                  215
       00241  0621 00 0 01001  10011         STA    *+1                                     216
       00242  0500 00 0 00000  10000         CLA    **                                      217
       00243  0601 60 0 00462  10001         STO*   ADRINT       ADRINT HAS A TAG OF 1      218
       00244  0500 00 0 00473  10001         CLA    TEMPXR                                  219
       00245  0601 60 0 41003  10011         STO*   *-3                                     220
       00246  2 00001 7 00217  10001  RWLPED TIX    RWSRCH-2,7,1 LOOP FOR N ROWS.           221
                                      *                                                     222
                                      *             NOW INTERCHANGE ROWS ACCORDING TO COLUMN TABLE  223
                                      *                                                     224
       00247  0534 00 7 00452  10001         LXA    N,7                                     225
       00250  0754 00 7 00000  10000         PXA    0,7                                     226
       00251  0734 00 1 00000  10000         PAX    0,1                                     227
       00252  0754 00 7 00000  10000  CLSRCH PXA    0,7                                     228

BINARY CARD ID. INV40011
       00253  0402 60 0 00463  10001         SUB*   ADCINT       ADCINT HAS A TAG OF 1      229
       00254  0100 00 0 00257  10001         TZE    FOUNDC                                  230
       00255  2 00001 1 00252  10001         TIX    CLSRCH,1,1                              231
       00256  0020 00 0 00431  10001         TRA    ERR4         CANNOT FIND COLUMN         232
       00257  0754 00 7 00000  10000  FOUNDC PXA    0,7                                     233
       00260  0634 00 1 00471  10001         SXA    TEMP1,1                                 234
       00261  0402 00 0 00471  10001         SUB    TEMP1                                   235
       00262  0100 00 0 00277  10001         TZE    CLLPED       ROW IS IN CORRECT PLACE    236
       00263  0634 00 7 00473  10001         SXA    TEMPXR,7                                237
       00264  0074 00 4 00476  10001         TSX    INRCHR,4                                238
237    00265  0074 00 0 00473  10001         TSX    TEMPXR                                  239
238    00266  0074 00 0 00471  10001         TSX    TEMP1                                   240
                                      *             INTERCHANGE COLUMN INTEGERS             241
       00267  0534 00 7 00473  10001         LXA    TEMPXR,7                                242
       00270  0500 00 0 00463  10001         CLA    ADCINT                                  243
       00271  0402 00 0 00473  10001         SUB    TEMPXR                                  244
       00272  0621 00 0 01001  10011         STA    *+1                                     245
       00273  0500 00 0 00000  10000         CLA    **                                      246
       00274  0601 60 0 00463  10001         STO*   ADCINT       ADCINT HAS A TAG OF 1      247
       00275  0500 00 0 00473  10001         CLA    TEMPXR                                  249

BINARY CARD ID. INV40012
       00276  0601 60 0 41003  10011         STO*   *-3                                     249
```

```
00277  2 00001 7 00250  10001  CLLPED TIX    CLSRCH-2,7,1  LOOP FOR N COLUMNS
                               *
                               *             INVERSE IS NOW IN THE CORRECT FORM - TIME TO
                               *             RESCALE THE SCALED INVERSE
                               *
00300  0500 00 0 00466  10001         CLA    CHRCNG
00301  0100 00 0 00361  10001         TZE    DSLPED+1      IF CHRCNG=0, RESCALING UNNECESSARY
00302  4120 00 0 00315  10001         TMI    RESCAL        IF NEGATIVE, NO OVERFLOW PROBLEMS
00303  0774 00 1 00001  10000         AXT    1,1           SET UP XR1 FOR FINDLE
00304  0500 00 0 00454  10001         CLA    ADA           SET UP TO PIVOT ROW 1 FOR FINDLE
00305  0400 00 0 00452  10001         ADD    N
00306  0621 00 0 00456  10001         STA    ADPVRW
00307  0074 00 4 00546  10001         TSX    FINDLE,4      FIND LARGEST ELEMENT IN SCALED INVERSE
00310  0500 00 0 00466  10001         CLA    CHRCNG
00311  0401 00 0 00471  10001         ADM    TEMP1         FINDLE STORES LARGEST IN TEMP1
00312  4760 00 0 00001  10000         PBT                  WAS THERE OVERFLOW
00313  0020 00 0 01002  10011         TRA    *+2           NO
00314  0020 00 0 00431  10001         TRA    ERR4          YES
00315  0534 00 7 00453  10001  RESCAL LXA    NSQ,7
00316  0500 00 0 00466  10001         CLA    CHRCNG
00317  0401 60 0 00461  10001         ADM*   APLNSQ        APLNSQ HAS A TAG OF 7
00320  0120 00 0 01003  10011         TPL    *+3           NO UNDERFLOW IF POSITIVE

BINARY CARD ID. INV40013
00321  0600 60 0 00461  10001         STZ*   APLNSQ        UNDERFLOW, STORE ZERO
00322  0020 00 0 00326  10001         TRA    RSCLED
00323  0560 60 0 00461  10001         LDQ*   APLNSQ
00324  0763 00 0 00000  10000         LLS    0
00325  0601 60 0 00461  10001         STO*   APLNSQ
00326  2 00001 7 00316  10001  RSCLED TIX    RESCAL+1,7,1  LOOP FOR 2*N**2 NUMBERS
                               *
                               *             NOW RESCALE THE DETERMINANT
                               *
00327  4520 60 0 00446  10001         NZT*   DET           IS MATRIX SINGULAR
00330  0020 00 0 00361  10001         TRA    DSLPED+1      IF SINGULAR, RETURN
00331  0500 00 0 00466  10001         CLA    CHRCNG
00332  0760 00 0 00002  10000         CHS
00333  0601 00 0 00466  10001         STO    CHRCNG
00334  0534 00 7 00452  10001         LXA    N,7
00335  0500 00 0 00466  10001  STDSCL CLA    CHRCNG
00336  0401 60 0 00446  10001         ADM*   DET
00337  4760 00 0 00001  10000         PBT
00340  0020 00 0 01002  10011         TRA    *+2           NO OVERFLOW
00341  0020 00 0 00350  10001         TRA    DIVDET        OVERFLOW, DIVIDE BY 10
00342  4120 00 0 00354  10001         TMI    MLTDET        UNDERFLOW, MULTIPLY BY 10
00343  0560 60 0 00446  10001         LDQ*   DET

BINARY CARD ID. INV40014
00344  0763 00 0 00000  10000         LLS    0
00345  0074 00 4 00654  10001         TSX    SCALE,4
00346  0601 60 0 00446  10001         STO*   DET
00347  0020 00 0 00360  10001         TRA    DSLPED        GO TO END OF LOOP
00350  0500 60 0 00446  10001  DIVDET CLA*   DET           DIVIDE DET BY TEN
00351  0241 60 0 00450  10001         FDP*   TEN
```

```
      00352  4600 60 0 00446  10001         STO*  DET
      00353  0020 00 0 00335  10001         TRA   STDSCL
      00354  0560 60 0 00446  10001  MLTDET LDQ*  DET          MULTIPLY DET BY TEN
      00355  0260 60 0 00450  10001         FMP*  TEN
      00356  0601 60 0 00446  10001         STO*  DET
      00357  0020 00 0 00335  10001         TRA   STDSCL
      00360  2 00001 7 00335  10001  DSLPED TIX   STDSCL,7,1   LOOP N TIMES FOR COMPLETE SCALING
                                     *
                                     *            END OF ALL ARITHMETIC NOW TIME TO ARRANGE
                                     *            ERROR CODES AND SUCH THEN RETURN
      00361  0774 00 1 00000  10000     XR1 AXT   **,1
      00362  0774 00 2 00000  10000     XR2 AXT   **,2
      00363  0774 00 3 00000  10000     XR3 AXT   **,3
      00364  0774 00 4 00000  10000     XR4 AXT   **,4
      00365  0774 00 5 00000  10000     XR5 AXT   **,5
      00366  0774 00 6 00000  10000     XR6 AXT   **,6

BINARY CARD ID. INV40015
      00367  0774 00 7 00000  10000     XR7 AXT   **,7
      00370  4500 00 0 00443  10001         CAL   TEMP8        RESTORE CELL 8
      00371  0602 00 0 00010  10000         SLW   8
      00372  0500 60 0 00446  10001         CLA*  DET
      00373  0601 60 4 00010  10000         STO*  8,4
      00374  0500 00 0 00451  10001         CLA   DETXP
      00375  0601 60 4 00011  10000         STO*  9,4
      00376  0500 00 0 00467  10001         CLA   ERRCD1
      00377  0601 60 4 00005  10000         STO*  5,4
      00400  0500 00 0 00470  10001         CLA   ERRCD2
      00401  0601 60 4 00006  10000         STO*  6,4
      00402  0020 00 4 00001  10000         TRA   1,4
                                     *
                                     *            FOLLOWING ARE ALL THE ERROR STOPS
                                     *
      00403  0500 00 0 00733  10001    ERR1 CLA   =1           OVERFLOW RETURN
      00404  0601 00 0 00467  10001         STO   ERRCD1
      00405  0634 00 1 00470  10001         SXA   ERRCD2,1     SAVE REDUCTION STAGE
      00406  0600 60 0 00446  10001         STZ*  DET          ZERO DETERMINANT
      00407  0600 00 0 00451  10001         STZ   DETXP
      00410  0020 00 0 00215  10001         TRA   REDOVR       TRANSFER BACK TO INTERCHANGE ROWS
                                     *                         AND COLUMNS, AND RESCALE
      00411  0500 00 0 00740  10001    ERR2 CLA   =2

BINARY CARD ID. INV40016
      00412  0601 00 0 00467  10001         STO   ERRCD1
      00413  0754 00 1 00000  10000         PXA   0,1
      00414  0402 00 0 00733  10001         SUB   =1
      00415  0601 00 0 00470  10001         STO   ERRCD2       RANK OF MATRIX =XR1-1
      00416  0020 00 0 00406  10001         TRA   ERR1+3       TRANSFER BACK TO INTERCHANGE
                                     *                         ROWS AND COLUMNS, AND RESCALE
      00417  0500 00 0 00467  10001    ERR3 CLA   ERRCD1
      00420  4100 00 0 00437  10001         TNZ   ERR3PL
      00421  0500 00 0 00741  10001         CLA   =3
      00422  0601 00 0 00467  10001         STO   ERRCD1
      00423  0600 00 0 00470  10001         STZ   ERRCD2
```

```
00424  0500 00 0 00734  10001         CLA   =1.           =0201400000000
00425  0400 00 0 00466  10001         ADD   CHRCNG
00426  0534 00 4 00364  10001         LXA   XR4,4
00427  0601 60 4 00007  10000         STO*  7,4           STORE SCALING FACTOR
00430  0020 00 0 00327  10001         TRA   RSCLED+1      GO BACK AND SCALE DETERMINANT
00431  0500 00 0 00467  10001   ERR4  CLA   ERRCD1        CANNOT RE-ARRANGE ROWS,COLUMNS -MACH. ERROR
00432  4100 00 0 00441  10001         TNZ   ERR4PL
00433  0500 00 0 00742  10001         CLA   =4
00434  0601 00 0 00467  10001         STO   ERRCD1        CANNOT RE-ORDER ROWS AND COLUMNS

BINARY CARD ID. INV40017
00435  0600 00 0 00470  10001         STZ   ERRCD2        FPT GIVES STORAGE TRAP
00436  0020 00 0 00361  10001         TRA   DSLPED+1      RETURN IMMEDIATELY
00437  0400 00 0 00743  10001 ERR3PL  ADD   =10
00440  0020 00 0 00422  10001         TRA   ERR3+3
00441  0400 00 0 00744  10001 ERR4PL  ADD   =20
00442  0020 00 0 00434  10001         TRA   ERR4+3
                              *
                              *             STORAGE FOR PROGRAM CONSTANTS
                              *
00443  0 00000 0 00000  10000  TEMP8  PZE   0             STORAGE FOR CORE LOCATION 8 (DEC)
00444  0020 00 0 00707  10001 FPTTRA  TRA   FLPSPL
00445  0 00000 0 00000  10000  CELLO  PZE   0             STORAGE FOR FLPSPL CODE
00446  0 00000 0 00000  10000    DET  PZE   0             ADDRESS OF DETERMINANT
00447  0 00000 0 00000  10000 MULTPL  PZE   0             ADDRESS OF REDUCTION FACTOR
00450  0 00000 0 00000  10000    TEN  PZE   0             ADDRESS OF FLOATING POINT 10.
00451  0 00000 0 00000  10000  DETXP  PZE   0             CELL FOR DETERMINANT EXPONENT
00452  0 00000 0 00000  10000      N  PZE   0
00453  0 00000 0 00000  10000    NSQ  PZE   0             N**2
00454  0 00000 0 00000  10000    ADA  PZE   0             ADDRESS OF MATRIX
00455  0 00000 0 00000  10000 ADPVEL  PZE   0             ADDRESS OF CURRENT PIVOT ELEMENT
00456  0 00000 7 00000  10000 ADPVRW  PZE   0,7           ADDRESS OF CURRENT PIVOT ROW +2N
00457  0 00000 6 00000  10000 ADDPEL  PZE   0,6           ADDRESS OF ELEMENT BEING REDUCED

BINARY CARD ID. INV40018
00460  0 00000 7 00000  10000 ADDPRW  PZE   0,7           ROW OF ADDPEL
00461  0 00000 7 00000  10000 APLNSQ  PZE   0,7           ADDRESS OF FIRST CELL BEYOND MATRIX
00462  0 00000 1 00000  10000 ADRINT  PZE   0,1           ADDRESS OF ROW INTEGERS + N
00463  0 00000 1 00000  10000 ADCINT  PZE   0,1           ADDRESS OF COLUMN INTEGERS + N
00464  0 00000 0 00000  10000 LGSTRW  PZE   0             ROW INDEX OF LARGEST ELEMENT IN SUB-MATRIX
00465  0 00000 0 00000  10000 LGSTCL  PZE   0             COLUMN INDEX OF LARGEST ELEM. IN SUB-MATRIX
00466  0 00000 0 00000  10000 CHRCNG  PZE   0             CHARACTERISTIC SCALE FACTOR
00467  0 00000 0 00000  10000 ERRCD1  PZE   0             ERROR CODE 1
00470  0 00000 0 00000  10000 ERRCD2  PZE   0             ERROR CODE 2
00471  0 00000 0 00000  10000  TEMP1  PZE   0
00472  0 00000 0 00000  10000  TEMP2  PZE   0
00473  0 00000 0 00000  10000 TEMPXR  PZE   0             TEMPORARY FOR ADDRESS ONLY
00474  0 00000 0 00000  10000 TEMPAC  PZE   0             TEMPORARY STORAGE FOR FLPSPL
00475  0 00000 0 00000  10000 TEMPMQ  PZE   0             SAME AS ABOVE
                              *             CLOSED SUBROUTINE TO INTERCHANGE ROWS
                              *             TWO ARGUMENTS -I,J=ROWS TO BE MOVED
                              *
                              *
00476  0560 60 4 00001  10000 INPCHR  LDQ*  1,4
```

```
      00477  0200 00 0 00452  10001          MPY    N
      00500  0131 00 0 00000  10000          XCA
      00501  0400 00 0 00454  10001          ADD    ADA            ADDRESS OF ROW I +  N
      00502  0621 00 0 00513  10001          STA    INR1

BINARY CARD ID. INV40019
      00503  0621 00 0 00516  10001          STA    INR1+3
      00504  0560 60 4 00002  10000          LDQ*   2,4
      00505  0200 00 0 00452  10001          MPY    N
      00506  0131 00 0 00000  10000          XCA
      00507  0400 00 0 00454  10001          ADD    ADA            ADDRESS OF ROW I +   N
      00510  0621 00 0 00514  10001          STA    INR1+1
      00511  0621 00 0 00515  10001          STA    INR1+2
      00512  0534 00 7 00452  10001          LXA    N,7
      00513  0500 00 7 00000  10000     INR1 CLA    **,7           ROW I
      00514  0560 00 7 00000  10000          LDQ    **,7           ROW J
      00515  0601 00 7 00000  10000          STO    **,7           ROW J
      00516  4600 00 7 00000  10000          STQ    **,7           ROW I
      00517  2 00001 7 00513  10001          TIX    INR1,7,1
      00520  0020 00 4 00003  10000          TRA    3,4            DONE
                                      *
                                      *         CLOSED SUBROUTINE TO INTERCHANGE COLUMNS
                                      *         TWO ARGUMENTS -I,J = COLUMNS TO BE MOVED
                                      *
      00521  0500 00 0 00454  10001   INRCHC CLA    ADA
      00522  0402 00 0 00733  10001          SUB    =1
      00523  0400 00 0 00453  10001          ADD    NSQ
      00524  0601 00 0 00472  10001          STO    TEMP2          A-1+  N**2
      00525  0500 60 4 00001  10000          CLA*   1,4

BINARY CARD ID. INV40020
      00526  0400 00 0 00472  10001          ADD    TEMP2          ADDRESS OF COLUMN I+ N**2+1
      00527  0621 00 0 00540  10001          STA    INC1
      00530  0621 00 0 00543  10001          STA    INC1+3
      00531  0500 60 4 00002  10000          CLA*   2,4
      00532  0400 00 0 00472  10001          ADD    TEMP2          ADDRESS OF COLUMN J +  N**2+1
      00533  0621 00 0 00541  10001          STA    INC1+1
      00534  0621 00 0 00542  10001          STA    INC1+2
      00535  0534 00 7 00453  10001          LXA    NSQ,7
      00536  0534 00 6 00452  10001          LXA    N,6
      00537  4634 00 6 00544  10001          SXD    INC2,6
      00540  0500 00 7 00000  10000     INC1 CLA    **,7           COLUMN I
      00541  0560 00 7 00000  10000          LDQ    **,7           COLUMN J
      00542  0601 00 7 00000  10000          STO    **,7           COLUMN J
      00543  4600 00 7 00000  10000          STQ    **,7           COLUMN I
      00544  2 00000 7 00540  10001     INC2 TIX    INC1,7,0       MAIN LOOP -2N IN DECREMENT
      00545  0020 00 4 00003  10000          TRA    3,4            DONE
                                      *
                                      *         CLOSED SUBROUTINE TO FIND THE LARGEST
                                      *         MAGNITUDE IN THE N-XR1+1 SQUARE SUB-MATRIX
                                      *         CURRENT REDUCTION STAGE IS IN XR1
                                      *         ROW LOCATION OF L.F. = LGSTRW
                                      *         COLUMN LOCATION OF L.F. = LGSTCL
                                      *
```

```
       00546  0754 00 1 00000  10000  FINDLE PXA   0,1
       00547  0402 00 0 00452  10001         SUB   N
       00550  0402 00 0 00733  10001         SUB   =1

BINARY CARD ID. INV40021
       00551  0734 00 3 00000  10000         PAX   0,3            COUNT OF ROWS IN XR3=N+1-XR1
       00552  0621 00 0 00473  10001         STA   TEMPXR         TEMPXR=2(N+1-XR1)-TO GO INTO XR7
       00553  0634 00 1 00464  10001         SXA   LGSTRW,1       START LOCATION AT PIVOT ELEMENT
       00554  0634 00 1 00465  10001         SXA   LGSTCL,1
       00555  0500 00 0 00456  10001         CLA   ADPVRW
       00556  0621 00 0 00562  10001         STA   FLE1+1
       00557  0600 00 0 00471  10001         STZ   TEMP1          KEEP LARGEST IN TEMP1
       00560  0534 00 7 00473  10001         LXA   TEMPXR,7
       00561  0754 00 0 00000  10000    FLE1 PXA   0,0
       00562  0401 00 7 00000  10000         ADM   **,7
       00563  0340 00 0 00471  10001         CAS   TEMP1
       00564  0020 00 0 01003  10011         TRA   *+3            GREATER - EXCHANGE
       00565  0020 00 0 01005  10011         TRA   *+5
       00566  0020 00 0 01004  10011         TRA   *+4
       00567  0601 00 0 00471  10001         STO   TEMP1
       00570  0634 00 3 00464  10001         SXA   LGSTRW,3
       00571  0634 00 7 00465  10001         SXA   LGSTCL,7
       00572  2 00001 7 00561  10001         TIX   FLE1,7,1       LOOP BACK IF ROW NOT DONE
       00573  4500 00 0 00562  10001         CAL   FLE1+1

BINARY CARD ID. INV40022
       00574  0400 00 0 00452  10001         ADD   N
       00575  0621 00 0 00562  10001         STA   FLE1+1         INCREMENT TO NEW-ROW
       00576  2 00001 3 00560  10001         TIX   FLE1-1,3,1     LOOP BACK IF MORE ROWS
       00577  4520 00 0 00471  10001         NZT   TEMP1          DONE, IS LARGEST ELEMENT = 0
       00600  0020 00 0 00411  10001         TRA   ERR2           SINGULAR MATRIX ERROR RETURN
                                      *                           ADJUST LGSTRW, CL
       00601  0502 00 0 00464  10001         CLS   LGSTRW
       00602  0400 00 0 00452  10001         ADD   N
       00603  0400 00 0 00733  10001         ADD   =1
       00604  0621 00 0 00464  10001         STA   LGSTRW
       00605  0502 00 0 00465  10001         CLS   LGSTCL
       00606  0400 00 0 00452  10001         ADD   N
       00607  0400 00 0 00733  10001         ADD   =1
       00610  0621 00 0 00465  10001         STA   LGSTCL
       00611  0020 00 4 00001  10000         TRA   1,4
                                      *
                                      *            CLOSED SUBROUTINE TO PERFORM ONE REDUCTION
                                      *            STAGE - ONE STAGE CONSISTS OF PRODUCING
                                      *            ZEROS ON THE PIVOT COLUMN AND A 1 IN THE
                                      *            PIVOT POSITION- ZEROS AND 1 ARE ONLY UNDERSTOOD
                                      *            DUE TO STORAGE OVER ITSELF
                                      *            XR1=REDUCTION STAGE
                                      *
       00612  0534 00 6 00453  10001  REDUCE LXA   NSQ,6
       00613  0634 00 1 00473  10001         SXA   TEMPXR,1
       00614  0500 00 0 00454  10001         CLA   ADA
       00615  0400 00 0 00452  10001         ADD   N
       00616  0621 00 0 00460  10001         STA   ADDPRW           N + ADDRESS OF ROW BEING REDUCED
```

INV4S
ASSEMBLED TEXT.

08/17/65

```
BINARY CARD ID. INV40023
     00617  0402 00 0 00452  10001          SUB   N
     00620  0400 00 0 00473  10001          ADD   TEMPXR
     00621  0402 00 0 00733  10001          SUB   =1
     00622  0400 00 0 00453  10001          ADD   NSQ          N**2 + ADDRESS OF ELEMENT BEING ZEROED
     00623  0621 00 0 00457  10001          STA   ADDPEL
     00624  0774 00 5 00001  10000          AXT   1,5          COUNT OF ROWS BEING REDUCED
     00625  0754 00 5 00000  10000  REDUC1  PXA   0,5
     00626  0402 00 0 00473  10001          SUB   TEMPXR
     00627  0100 00 0 00646  10001          TZE   REDUC3       DO NOT OPERATE ON PIVOT ROW
     00630  0500 60 0 00457  10001          CLA*  ADDPEL       ADDPEL HAS A TAG OF 6
     00631  0100 00 0 00646  10001          TZE   REDUC3       IF ALREADY ZERO, NO REDUCTION NECESSARY
     00632  0241 60 0 00455  10001          FDP*  ADPVEL
     00633  0131 00 0 00000  10000          XCA
     00634  0760 00 0 00002  10000          CHS                USE NEGATIVE RATIO
     00635  0601 60 0 00447  10001          STO*  MULTPL       REDUCTION FACTOR
     00636  0534 00 7 00452  10001          LXA   N,7
     00637  0560 60 0 00456  10001  REDUC2  LDQ*  ADPVRW       ADPVRW HAS A TAG OF 7
     00640  0260 60 0 00447  10001          FMP*  MULTPL
     00641  0300 60 0 00460  10001          FAD*  ADDPRW       ADDPRW HAS A TAG OF 7

BINARY CARD ID. INV40024
     00642  0601 60 0 00460  10001          STO*  ADDPRW
     00643  2 00001 7 00637  10001          TIX   REDUC2,7,1
     00644  0500 60 0 00447  10001          CLA*  MULTPL       PUT REDUCTION FACTOR WHERE
     00645  0601 60 0 00457  10001          STO*  ADDPEL       ZERO WAS PRODUCED
     00646  0500 00 0 00460  10001  REDUC3  CLA   ADDPRW
     00647  0400 00 0 00452  10001          ADD   N
     00650  0621 00 0 00460  10001          STA   ADDPRW       GO TO NEXT ROW
     00651  1 00001 5 01001  10011          TXI   *+1,5,1      XR5 = ROW ABOUT TO BE REDUCED
     00652  2 00000 6 00625  10001  REDUC4  TIX   REDUC1,6,**   N IN DECREMENT
     00653  0020 00 4 00001  10000          TRA   1,4          DONE
                                    *
                                    *             CLOSED SUBROUTINE TO SCALE AN INCOMING
                                    *             NUMBER SO THAT IT IS LESS THAN 10. AND
                                    *             GREATER THAN OR EQUAL TO 1. DIVISIONS AND
                                    *             MULTIPLICATIONS KEPT TRACK OF IN DETXP.
                                    *             NUMBER EXPECTED IN AC AND NO-RESULT LEFT THERE
                                    *
     00654  0601 00 0 00474  10001  SCALE   STO   TEMPAC
     00655  0760 00 0 00003  10000          SSP
     00656  0340 60 0 00450  10001  SCALE1  CAS*  TEN
     00657  0020 00 0 00674  10001          TRA   DIVIDE
     00660  0020 00 0 00674  10001          TRA   DIVIDE
     00661  0340 00 0 00734  10001  TSTONE  CAS   =1.
     00662  0020 00 0 00703  10001          TRA   CHKSGN
     00663  0020 00 0 00703  10001          TRA   CHKSGN
     00664  0131 00 0 00000  10000          XCA

BINARY CARD ID. INV40025
     00665  0260 60 0 00450  10001          FMP*  TEN
     00666  0601 60 0 00447  10001          STO*  MULTPL       TEMPORARY STORAGE
     00667  0500 00 0 00451  10001          CLA   DETXP
     00670  0402 00 0 00733  10001          SUB   =1
```

```
      INV4S                                                     08/17/65
    ASSEMBLED TEXT.

      00671  0601 00 0 00451  10001         STD    DETXP
      00672  0500 60 0 00447  10001         CLA*   MULTPL
      00673  0020 00 0 00661  10001         TRA    TSTONE
      00674  0241 60 0 00450  10001  DIVIDE FDP*   TEN
      00675  4600 60 0 00447  10001         STQ*   MULTPL
      00676  0500 00 0 00451  10001         CLA    DETXP
      00677  0400 00 0 00733  10001         ADD    =1
      00700  0601 00 0 00451  10001         STO    DETXP
      00701  0500 60 0 00447  10001         CLA*   MULTPL
      00702  0020 00 0 00656  10001         TRA    SCALE1
      00703  0560 00 0 00474  10001  CHKSGN LDQ    TEMPAC
      00704  0763 00 0 00000  10000         LLS    0
      00705  0560 00 0 00745  10001         LDQ    =0
      00706  0020 00 4 00001  10000         TRA    1,4
                                     *
                                     *             FLOATING POINT SPILL ROUTINE TO ANALYZE
                                     *             OVER/UNDER FLOW DURING DOUBLE PRECISION
                                     *             OPERATIONS
                                     *
                                     *
                                     *
                                     *
      00707  0601 00 0 00474  10001  FLPSPL STO    TEMPAC

BINARY CARD ID. INV40026
      00710  0500 00 0 00000  10000         CLA    0
      00711  0622 00 0 00445  10001         STD    CELLO
      00712  0500 00 0 00445  10001         CLA    CELLO
      00713  0340 00 0 00746  10001         CAS    =3817
      00714  0020 00 0 00723  10001         TRA    GRT3
      00715  0020 00 0 00721  10001         TRA    ALLUND
      00716  0500 00 0 00474  10001         CLA    TEMPAC
      00717  0560 00 0 00745  10001         LDQ    =0
      00720  0020 60 0 00000  10000         TRA*   0              RETURN
      00721  0754 00 0 00000  10000  ALLUND PXA    0,0
      00722  0020 00 0 41003  10011         TRA    *-3
      00723  0340 00 0 00747  10001    GRT3 CAS    =9817
      00724  0020 00 0 01003  10011         TRA    *+3
      00725  0020 60 0 00000  10000         TRA*   0
      00726  0020 00 0 00403  10001         TRA    ERR1
      00727  0340 00 0 00750  10001         CAS    =11817
      00730  0020 00 0 00403  10001         TRA    ERR1
      00731  0020 00 0 00721  10001         TRA    ALLUND
      00732  0020 00 0 00721  10001         TRA    ALLUND

BINARY CARD ID. INV40027
      00733  000000000001     10000         *LORG
      00734  201400000000     10000
      00735  204500000000     10000
      00736  377000000000     10000
      00737  177000000000     10000
      00740  000000000002     10000
      00741  000000000003     10000
      00742  000000000004     10000
```

```
        INV4S                                                          08/17/65
      ASSEMBLED TEXT.

    00743  000000000012      10000
    00744  000000000024      10000
    00745  000000000000      10000
    00746  000003000000      10000
    00747  000011000000      10000
    00750  000013000000      10000
                       00000  01111         END                                                   591



        INV4S                                                          08/17/65
      CONTROL DICTIONARY

            $CDICT  INV4S                                                                INV40028

BINARY CARD ID. INV40029
            000751000000                PREFACE      START=0,LENGTH=489,TYPE=7094,CMPLX=5
            000004000005
            314565046260        INV4S   DECK         LOC=0,LENGTH=489
            000751000000
            314565046260        INV4S   REAL         LOC=0,LENGTH=0
            000000000000

            $DKEND  INV4S                                                                INV40030
```

```
                                                                      05/22/67
        INVBB      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

       SUBROUTINE INVBB

C      FOR WING BODY CASE, PREPARES AERODYNAMIC INFLUENCE MATRIX FOR
C      MATRIX REDUCTION
C      MAXIMUM SIZE MATRIX INVERSION =110

       COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
      1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

       DIMENSION ABB(115,115),AWB(110)

       MDEMEN=115

C      READ AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY
C      AND INFLUENCE ON WING DUE TO BODY) INTO CORE

       CALL FSF(1,NTAPEA,IRR)                                                  3
       DO 100 J=1,NBODY
       READ (NTAPEA) (ABB(I,J),I=1,NBODY),(AWB(I  ),I=1,NWING)

                                                                               7
       WRITE (NTAPED) (AWB(I),I=1,NWING)                                      16
  100  CONTINUE
       END FILE NTAPED
C      INVERT AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY)
                                                                              23
       CALL SINVRT(ABB,MDEMEN,NBODY,IRR1,IRR2,SCALE,DET,NDETXP)               24
       IF (IRR1) 150,200,150
  150  CONTINUE
       WRITE (NTAPED,6000) IRR1,IRR2,SCALE                                    27
       REWIND NTAPEA                                                          28
       STOP


  200  CONTINUE
       DO 300 J=1,NBODY
       WRITE (NTAPED) (ABB(I,J),I=1,NBODY)                                    32
  300  CONTINUE
       END FILE NTAPED                                                        39
       REWIND NTAPED

                                                                              40
 6000  FORMAT(1H1,33HERROR IN INVERSION OF BODY MATRIX,5X,
      16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

       RETURN
       END
```

```
      SUBROUTINE INVRW

C     INVERT REDUCED MATRIX
C     MAXIMUM SIZE MATRIX INVERSION =110

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF
     1,NTAPEI,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION ARWW(115,115)

      MDEMEN=115
      DO 700 J=1,NWING
      READ (NTAPEE) (ARWW(I,J),I=1,NWING)                          5
  700 CONTINUE
      END FILE NTAPEE

C     MATRIX INVERSION                                             12
      CALL SINVRT(ARWW,MDEMEN,NWING,IRR1,IRR2,SCALE,DET,NDETXP)    13
      IF (IRR1) 750,800,750
  750 CONTINUE
      WRITE (NTAPEO,6010) IRR1,IRR2,SCALE                          16
      REWIND NTAPEE                                                17
      STOP

  800 CONTINUE
      DO 850 J=1,NWING
      WRITE (NTAPEE) (ARWW(I,J),I=1,NWING)                         21
  850 CONTINUE
      END FILE NTAPEE                                              28
      REWIND NTAPEE
                                                                   29
 6010 FORMAT(1H1,36HERROR IN INVERSION OF REDUCED MATRIX,5X,
     16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN
      END
```

```
      SUBROUTINE INVW
C     FOR WING ONLY CASE, STORES MATRIX (A) AND (A) INVERSE ON TAPE
C     MAXIMUM SIZE MATRIX INVERSION =110


      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION AWW(115,115)

      MCEMEN=115

C     READ (A) MATRIX INTO CORE, WRITE ON TAPE

      CALL FSF(1,NTAPEA,IRR9                                            3
      DO 100 J=1,NWING
      READ (NTAPEA) (AWW(I,J),I=1,NWING)                                7
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)                               12
  100 CONTINUE
      END FILE NTAPEE                                                   19
      REWIND NTAPEA

C     INVERT MATRIX (A)
                                                                        20
      CALL SINVRT(AWW,MDEMEN,NWING,IRR1,IRR2,SCALE,DET,NDETXP)          21
      IF (IRR1) 150,200,150
  150 WRITE (NTAPEO,6000) IRR1,IRR2,SCALE                               24
      REWIND NTAPEE                                                     25
      STOP


  200 CONTINUE
      DO 250 J=1,NWING
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)                               29
  250 CONTINUE
      END FILE NTAPEE                                                   36
      REWIND NTAPEE
                                                                        37
 6000 FORMAT(1H1,38HERROR IN INVERSION OF WING ONLY MATRIX
     1,5X,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN
      END
```

```
         IPACK                                                                 07/31/65
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

CIPACK      MICHAEL SYNGE           6-9247       COMPILED   3 SEP 64
C  TO PACK A BUFFER BY MOVING ALL FULL ARRAYS TO THE BEGINNING          IPAC0030
C  AND CONDENSING ALL THE EMPTY ARRAYS INTO ONE AT THE END.              IPAC0040
C                                                                       IPAC0060
      FUNCTION IPACK(BUF)                                               IPAC0070
      DIMENSION BUF(1)                                                  IPAC0080
C                                                                       IPAC0100
C              N      - TOTAL LENGTH OF BUFFER                          IPAC0110
C              M      - FIRST CELL OF CURRENT ARRAY                     IPAC0120
C              J      - TOTAL NUMBER OF FREE CELLS                      IPAC0130
C                                                                       IPAC0140
      IF(ICODE(BUF)-4)90,2,90                                           IPAC0150   ,1
    2 N=-IBFR(BUF(3))                                                   IPAC0160   ,2
      IF(N)80,80,3                                                      IPAC0170   ,3
    3 M=4                                                               IPAC0180   ,4
      J=0                                                               IPAC0190   ,5
    5 L=LENG(BUF(M-1))                                                  IPAC0200   ,6
      IF (ICODE(BUF(M-2)))10,40,10                                      IPAC0210   ,7
   10 IF (J)50,50,20                                                    IPAC0220
C                                                                       IPAC0230
C              MOVE UP FULL ARRAY                                       IPAC0240
C                                                                       IPAC0250   ,8
   20 I1=M-2-J                                                          IPAC0260   ,9
      I2=I1+L+1                                                         IPAC0270   ,10
      K=M-2                                                             IPAC0280   ,11
      DO 30I=I1,I2                                                      IPAC0290   ,12
      BUF(I)=BUF(K)                                                     IPAC0300   ,13
   30 K=K+1                                                             IPAC0310   ,14    ,15
      BUF(I1)=ARFUL(I1+2)                                               IPAC0320   ,16
      CALL UPDATE(BUF(I1+1))                                            IPAC0330   ,17
      GO TO 50                                                          IPAC0340   ,18
   40 J=J+L+2                                                           IPAC0350   ,19
   50 IF (N-L-M)100,100,70                                              IPAC0360   ,20
   70 M=M+L+2                                                           IPAC0370   ,21
      GO TO 5                                                           IPAC0380   ,22
  100 IF (J)1000,1000,110                                               IPAC0390
C                                                                       IPAC0400
C              SET UP FINAL (EMPTY) ARRAY                               IPAC0410
C                                                                       IPAC0420   ,23
  110 I1=N-J+1                                                          IPAC0430   ,24
      IPACK=J-2                                                         IPAC0440   ,25
      BUF(I1)=ARFRE(I1+2)                                               IPAC0450   ,26
      BUF(I1+1)=ARSEC(IPACK,IPACK)                                                 ,27
      GO TO 1000                                                        IPAC0470
C                                                                       IPAC0480
C              ERRORS                                                   IPAC0490
C                                                                       IPAC0500   ,28
   80 IPACK=-2                                                          IPAC0510   ,29
      GO TO 1000                                                        IPAC0520   ,30
   90 IPACK=-1                                                          IPAC0530   ,31
 1000 RETURN                                                            IPAC0540
C                                                                       IPAC0550   ,32
      END                                                               IPAC0560   ,33
```

```
          IRLEAS                                                                        07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

CIRLEAS      MICHAEL SYNGE             6-9247       COMPILED  21 SEP 64
C               TO RELEASE ALL EXCEPT THE                                        IRLE0030
C               FIRST K WORDS OF AN ARRAY                                        IRLE0040
C                                                                                IRLE0060
      FUNCTION IRLEAS(IRRAY,K)                                                   IRLE0070
      DIMENSION IRRAY(1)                                                         IRLE0080
C                                                                                IRLE0100
C               M       - INITIAL LENGTH OF THE ARRAY                            IRLE0110
C               K       - THE NUMBER OF WORDS TO BE RETAINED                     IRLE0120
C                                                                                IRLE0130
      N=K+2                                                                      IRLE0140   ,1
      IF(N-2)70,70,5                                                             IRLE0150   ,2
    5 I=32767                                                                    IRLE0160   ,3
      IF (ICODE(IRRAY(I)))90,50,10                                               IRLE0170   ,4
   10 M=LENG(IRRAY(I+1))                                                         IRLE0180   ,5
      IF (M-N)50,20,20                                                           IRLE0190   ,6
   20 IRRAY(I+1)=IRRAY(I+1)-262144*(M-N+2)                                                  ,7
      IRRAY(N-1)=IRFRE(N-IBFR(IRRAY))                                            IRLE0210   ,8
      IRRAY(N)=IRSEC(M-N,IRLEAS)                                                            ,9
   50 IRLEAS=0                                                                   IRLE0230   ,10
      GO TO 100                                                                  IRLE0240   ,11
   70 IRLEAS=IDLETE(IRRAY)                                                       IRLE0250   ,12
      GO TO 100                                                                  IRLE0260   ,13
   90 IRLEAS=-1                                                                  IRLE0270   ,14
  100 RETURN                                                                     IRLE0280
C                                                                                IRLE0290
C                                                                                IRLE0300   ,15
      END                                                                        IRLE0310   ,16
```

```
          IRSERV
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)              07/31/65

CIRSERV     MICHAEL SYNGE            6-9247       COMPILED  21 AUG 64
C  TO RESERVE AN ARRAY OF N CELLS IN A BUFFER. THE ARRAY CAN              IRSE0030
C  SUBSEQUENTLY BE REFERRED TO AS  BUF(NAME) .                            IRSE0040
C                                                                         IRSE0050
C                                                                         IRSE0070
      FUNCTION   IRSERV(N,BUF,NAME)                                       IRSE0080
C                                                                         IRSE0090
      M=-N                                                                IRSE0100    ,1
      IRSERV=IADARY(M,M,BUF,NAME)                                         IRSE0110    ,2
      RETURN                                                              IRSE0120
C                                                                         IRSE0130    ,3
      END                                                                 IRSE0140    ,4
```

```
                                                                         06/02/67
     KARMOR     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE KARMOR(NBODYS,NTHETA,      XN,YN,ZN,CPCALC,VOUT,ARA
     1,XP,ZP,RFAREA,XB,R,THETAB,ZDELTA,XC,YC,ZC,THETA,U,VV,WW,AN1,CPB
     2,BBCL,BBCD,BBCM)

      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE

      DIMENSION XB(1),R(1),THETAB(1),ZDELTA(1),XC(1),YC(1),ZC(1)
     1,U(1),VV(1),WW(1),AN1(1),CPB (55,1),THETA(1)
      DIMENSION ALPHAC(55),DTHETA(10),T(55),TC(55),TX(55),CPBB(10,55)
     1,UB(55,10),VB(55,10),VTB(55,10),SY(2)
      DIMENSION FD(55),SD(55),NBS(10),C(10)


C     CALCULATION OF VELOCITY COMPONENTS DUE TO BODY THICKNESS
C     USING SMOOTH BODY SOLUTION


      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))
                                                                      2     3     4

      PI=3.1415926
      XMACH2=XMACH*XMACH
      BT2=XMACH2-1.                                                   7
      BETA=SQRT(BT2)
      N1=NBODYS-1
      NACEL=0

      DO 60 J=1,N1
   60 ALPHAC(J)=ARA-(ZDELTA(J+1)-ZDELTA(J))/(XB(J+1)-XB(J))
      DO 65 J=1,NTHETA
   65 THETAB(J)=THETAB(J)/57.2957795

C     COMPUTE FIRST AND SECOND DERIVATIVES
      CALL DERIVS(XB,R,NBODYS,FD,SD)                                  29

      TANDEL=FD(1)
      COTDEL=1./FD(1)
      COB=COTDEL/BETA

C     CALCULATE SOURCE STRENGTHS
      NB=0
      DC 160 I=1,N1
      M=I+1
      UBSUM=0.
      VBSUM=0.
      UBDSUM=0.
      VBDSUM=0.
      VTDSUM=0.
      UC=0.
      VC=0.
      PHIX=0.
      PHISUM=0.
```

```
      PHID=0.

      RS=ALPHAC(I)
      RDRDXS=FD(I+1)
      TRMO=XB(I+1)-XB(1)+BETA*R(1)
      TRM2=BETA*R(I+1)
      TRM3=TRM2/TRMO
      TRM4 = SQRT(1.-TRM3*TRM3)                                      40
      RAB1=BETA*SQRT(1.-TRM3**2)/TRM3                                41
      RUB1=-ACOSH(TRMO/TRM2)
      RABD=-.5*BT2*(SQRT(1.-TRM3**2)/TRM3**2-RUB1)                   42
      RUBD=RAB1
      VTD=-.5*BT2*(SQRT(1.-TRM3**2)/TRM3**2+RUB1)                    43
      RN1=RAB1-FD(I+1)*RUB1
      RM1=RABD-FD(I+1)*RUBD
      IF(I.NE.1)GO TO 100
      IF(I.EQ.1.AND.R(1).NE.0.)GO TO 80

C     IF I EQUALS 1, DO TANGENT CONE SOLUTION
C     OR CORNER SOLUTION FOR NOSE
      T11= TANDEL/(SQRT(COTDEL**2-BT2)+TANDEL*ACOSH(COB))            52
      TC11 = -ALPHAC(1)/RM1
      GO TO 85
   80 T11=0.
      TC11=0.
   85 CONTINUE
      DO 87 K=1,NTHETA
      UB(1,K) = -ACOSH(COB)*T11+COS(THETAB(K))*RUBD*TC11             59
      VB(1,K) = SQRT(COTDEL**2-BT2)*T11+COS(THETAB(K))*RABD*TC11     61   63
   87 VTB(1,K) = SIN(THETAB(K))*VTD*TC11
                                                                     67
  100 CONTINUE
      DO 155 J=1,I
      NL=NB
      RNCC=0.

      TRM1=XB(I+1)-XB(J)+BETA*R(J)
      IF (R(I+1).EQ.0.) GO TO 152
      TRM3=TRM2/TRM1
      TRM32=TRM3**2
      TRM4=SQRT(1.-TRM3*TRM3)                                        81
      TRM5=ACOSH(TRM1/TRM2)
      RAB2=BETA*TRM1*(TRM4/TRM3-TRM3*TRM5)
      RUB2=2.*TRM1*(TRM5-TRM4)
      RN=RAB2+FD(I+1)*RUB2
      IF (TRM3.EQ.0.) RN=0.
      RABD2=-TRM1**3/R(I+1)**2*((1.-4.*TRM32)/3.*TRM4+TRM32*TRM5)
      RUBD2=TRM1**2/R(I+1)*(TRM4-TRM32*TRM5)
      VTD2=-TRM1**3/R(I+1)**2*((1.+2.*TRM32)/3.*TRM4-TRM32*TRM5)
      RM=RABD2-FD(I+1)*RUBD2
```

```
      IF (J .NE. I) GO TO 150

  130 T(J)=(RDRDXS-RN1*T11-RNCC)/RN

      TC(J)=-(RS+RM1*TC11)/RM

  150 RDRDXS=RDRDXS-RN*T(J)
      IF (J  .NE. 1) GO TO 151
      UBSUM = RUB1*T11
      VBSUM = RAB1*T11
      UBDSUM = RUBD*TC11
      VBDSUM = RABD*TC11
      VTDSUM = VTD*TC11
  151 CONTINUE
      UBSUM = UBSUM - RUB2*T(J)+UC
      VBSUM = VBSUM + RAB2*T(J)+VC
      UBDSUM = UBDSUM + RUBD2*TC(J)
      VBDSUM = VBDSUM + RABD2*TC(J)
      VTDSUM = VTDSUM + VTD2*TC(J)
      PHIX=PHIX-RUB2*T(J)
      RS=RS+RM*TC(J)
      GO TO 155
  152 IF(I.EQ.J) GO TO 154
      IF (J.NE.1) GO TO 153
      PHISUM = TRMO*T11
      PHID=TRMO*TC11
  153 PHISUM = PHISUM+TRM1*TRM1*T(J)
      PHID=PHID+TRM1*TRM1*TC(J)
      GO TO 155
                                                                 06/02/67
KARMOR     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

  154 T(J) = -PHISUM/TRM1**2
      TC(J)=-PHID/TRM1**2
  155 CONTINUE
      DO 157 K=1,NTHETA
      UB(M,K)=UBSUM+UBDSUM*COS(THETAB(K))                                       129
      VB(M,K)=VBSUM+VBDSUM*COS(THETAB(K))                                       132
  157 VTB(M,K)=VTDSUM*SIN(THETAB(K))                                            136
 160  CONTINUE

C     CALCULATION OF VELOCITY COMPONENTS IN THE FIELD
      IF (NWING) 700,800,700
  700 CONTINUE
      IF (NACEL.NE.0) GO TO 703
      NS=NBODY+1
      JM=1
      SY(1)=0.
      GO TO 705
  703 NS=1
      JM=2
      SY(1)=1.
      SY(2)=-1.
```

```
705 CONTINUE
      NPANEL=NBODY+NWING
      II=0
      DO 790 I=NS,NPANEL
      II=II+1
      U(II)=0.
      VV(II)=0.
      WW(II)=0.
      AN1(II)=0.
      DO 790 N=1,JM
      THETAA=ATN1(YC(I)-SY(N)*YN,ZC(I)-ZN)                          169
      COSTHA=COS(THETAA)                                              170
      SINTHA=SIN(THETAA)                                              171
      R2=(YC(I)-SY(N)*YN)**2+(ZC(I)-ZN)**2                           175
      R1=SQRT(R2)
      BR2=BT2*R2
      NC=0
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VT=0.
      VTF = 0.
      VF = 0.
      TRM2 = SQRT(BR2)                                                177
      DO 780 J=1,N1
      TRM1 = (XC(I)-XN)-XB(J)+BETA*R(J)
      TRM3 = TRM2/TRM1
      TRM32=TRM3*TRM3
      IF (TRM3.GT.1..OR.TRM3.LE.0.) GO TO 780
C     VELOCITY DUE TO LINEARLY VARING SINGULARITIES
      TRM4=SQRT(1.-TRM32)                                             186
      TRM5=ACOSH(TRM1/TRM2)
      US=-2.*TRM1*(TRM5-TRM4)*T(J)
      VS= BETA*TRM1*(TRM4/TRM3-TRM3*TRM5)*T(J)
      UD= TRM1**2/R1*(TRM4-TRM32*TRM5)*TC(J)
      VD= -TRM1**3/R1**2*((1.-4.*TRM32)/3.*TRM4+TRM32*TRM5)*TC(J)
      VT=-TRM1**3/R2*((1.+2.*TRM32)/3.*TRM4-TRM32*TRM5)*TC(J)
C     VELOCITY DUE TO TANGENT CONE
      IF (J.NE.1) GO TO 710
      US = US-ACOSH(TRM1/TRM2)*T11
      VS = VS+BETA*SQRT(1.-TRM32)/TRM3*T11                            195
      UD = UD+BETA*SQRT(1.-TRM32)/TRM3*TC11                           196
      VD = VD-.5*BT2*(SQRT(1.-TRM32)/TRM32-BETA*SQRT(1.-TRM32)/TRM3)
     $*TC11                                                           197
      VT = VT-.5*BT2*(SQRT(1.-TRM32)/TRM32-ACOSH(TRM1/TRM2))*TC11     198
  710 CONTINUE
```

```
C     SUMMATION OF VELOCITY COMPONENTS
  750 CONTINUE
      U(II) = U(II)+US+COSTHA*UD
      VF    = VF    +VS+COSTHA*VD
      VTF     = VTF     + VT*SINTHA
      TRM = (XC(I)-XN)-XB(NBODYS)+BETA*R(NBODYS)
      IF (TRM.LE.TRM2) GO TO 775
      XI = XB(NBODYS)-(XB(J)-BETA*R(J))-BETA*R(NBODYS)
      TRM6 = SQRT(TRM*TRM-BR2)                                            213
      IF (J.NE.1) GO TO 765
      U(II) = U(II)+T11*(ACOSH(TRM/TRM2)+XI/TRM6)
      VF     = VF  -T11*(TRM1*TRM-BR2)/(R1*TRM6)

  765 CONTINUE
      U(II) = U(II)-.5*(3.*TRM6+(TRM1*TRM1-2.*TRM1*XI-XI*XI-BR2)/TRM6-
     $4.*TRM1*ACOSH(TRM/TRM2))*T(J)
      VF     = VF    +(((TRM1*TRM1+BR2)*XI-TRM1*(TRM1*TRM1-BR2))/(R1*TRM6)
     $+BR2/R1*ACOSH(TRM/TRM2))*T(J)
  775 CONTINUE
  780 CONTINUE

      VV(II) = VF*SINTHA+VTF*COSTHA
      WW(II) = VF*COSTHA-VTF*SINTHA
      SYN=2*(JM-N)-1
      THETAS=THETAA-SYN*THETA(I)
      AN1(II) = VF*COS(THETAS)-VTF*SIN(THETAS)                            230   231

  790 CONTINUE
  800 CONTINUE

      NT=NTHETA-1
      DTHETA(1)=(THETAB(2)+THETAB(1))/2.
      DTHETA(NTHETA)=PI-(THETAB(NTHETA)+THETAB(NT))/2.
      DO 310 J=2,NT
  310 DTHETA(J)=(THETAB(J+1)-THETAB(J-1))/2.


C     CALCULATION OF PRESSURES, FORCES, AND MOMENTS
      BODYL=XB(NBODYS)-XB(1)
      DO 360 J=1,NTHETA
      DO 350 I=1,NBODYS
      IF (CPCALC-1.) 320,330,335
C     LINEAR CP CALCULATIONS
  320 CONTINUE
      CPBB(J,I)=-2.*UB(I,J)
      GO TO 340
C     NON-LINEAR CP CALCULATIONS
  330 CONTINUE
      XM2=XMACH2-1.
      CPBB(J,I)=-2.*UB(I,J)+XM2*UB(I,J)**2-VTB(I,J)**2-VB(I,J)**2
      GO TO 340
C     EXACT CP CALCULATIONS
```

```
335 CONTINUE                                                                         273
    UBO=COS(ARA)+UB(I,J)                                                             275   276
    VBO=SIN(ARA)*COS(THETAB(J))+VB(I,J)                                              279   280
    VTBO=VTB(I,J)-SIN(ARA)*SIN(THETAB(J))
    Q2=UBO**2+VBO**2+VTBO**2                                                         283
    CPBB(J,I)=1.42857/XMACH2*((1.+.2*XMACH2*(1.-Q2))**3.5-1.)
340 CONTINUE
350 CONTINUE
360 CONTINUE

    BBCL=0.
    BBCD=0.
    BBCM=0.
    CALL SCX(XB,R,FD,THETAB,CPBB,1,NBODYS,NTHETA,XP,ZP,CX,IRROR)                     291
    CALL SCN(XB,R,FD,THETAB,CPBB,1,NBODYS,NTHETA,XP,ZP,CN,IRROR)                     293

                                                                        06/02/67
         KARMOR     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

    CALL SCM(XB,R,FD,THETAB,CPBB,1,NBODYS,NTHETA,XP,ZP,CM,IRROR)                     295
    BBCL = BBCL + (CN*COS(ARA)-CX*SIN(ARA))                                          297   298
    BBCD = BBCD + (CN*SIN(ARA)+CX*COS(ARA))                                          299   300
    BBCM = BBCM - CM+CX*ZP

    NST=1
    IF (R(1).NE.0.) NST=2
    NBODYS=NBODYS-NST
    DC 585 I=NST,N1
    K=I-NST+1
    DO 580 J=1,NTHETA
580 CPB (K,J)=CPBB(J,I)
585 CONTINUE
    DO 590 J=1,NTHETA
590 THETAB(J)=THETAB(J)*57.2957795
```

```
C     OPTIONAL PRINT OF VELOCITY COMPONENTS
      WRITE (NTAPEO,6005) (XB(I),ZDELTA(I),R(I),FD(I),SD(I),I=1,NBODYS)          325
      IF (VOUT) 610,999,610
  610 CONTINUE
      WRITE (NTAPEO,6010)                                                        337
      WRITE (NTAPEO,6025)                                                        338
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                                 339
      WRITE (NTAPEO,6020)                                                        346
      DC 620 I=1,NBODYS
  620 WRITE (NTAPEO,6000) XB(I),(UB(I,J),J=1,NTHETA)                             350
      WRITE (NTAPEO,6030)                                                        357
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                                 358
      WRITE (NTAPEO,6020)                                                        366
      DO 630 I=1,NBODYS
  630 WRITE (NTAPEO,6000) XB(I),(VB(I,J),J=1,NTHETA)                             370
      WRITE (NTAPEO,6035)                                                        377
      WRITE (NTAPEO,6015) (THETAB(J),J=1,NTHETA)                                 378
      WRITE (NTAPEO,6020)                                                        385
      DO 640 I=1,NBODYS
  640 WRITE (NTAPEO,6000) XB(I),(VTB(I,J),J=1,NTHETA)
                                                                                 389
  999 CCNTINUE
      CALL FSF(9,NTAPEC,IRR)                                                     398
      IF (NACEL.EQ.1) GO TO 1000
      WRITE (NTAPEC) NBODYS                                                      402
      WRITE (NTAPEC) (XB(J),R(J),J=1,NBODYS)                                     403
      WRITE (NTAPEC) (T(J),TC(J),J=1,NBODYS)                                     411
      WRITE (NTAPEC) T11,TC11                                                    419
 1000 CONTINUE
      END FILE NTAPEC                                                            420
      REWIND NTAPEC                                                              421
      RETURN
 6000 FORMAT(1H ,F14.4,F15.5,9F10.5/(F29.5,9F10.5))
 6005 FORMAT (1H1//14H BODY GEOMETRY/56X,5HFIRST,9X,6HSECCND/9X,9HX-STAT
     $ICN,7X,6HCAMBER,9X,6HRADIUS,2X,2(5X,10HDERIVATIVE)//(1X,5F15.4))
 6010 FORMAT(1H1//66H VELOCITY COMPONENTS ON BODY DUE TO BODY LINE SOURC
     1ES AND DOUBLETS)
 6015 FORMAT(1H0,4X,11HTHETA(DEG.),F.14.4,9F10.4/(F29.4,9F10.4))
 6020 FORMAT(1H ,9X,1HX)
 6025 FCRMAT(1H0,5X,8HAXIAL(U))
 6030 FCRMAT(1H0,5X,10HRADIAL(VR))
 6035 FORMAT(1H0,5X,14HTANGENTIAL(VT))
       END
```

```
      SUBROUTINE LACKEY(ISTR,XB,YB,ZB,U,V,W,CP)
      COMMON /THICK/THKW,ARA,CPCALC,CAMN
      COMMON /FLOV1/KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM
      COMMON /FLOV4/COSARA,SINARA,XMACH2,BT2
      COMMON /FLOV5/N1,N2,N3,N4
      U=0.
      V=0.
      W=0.
      UT=0.
      VT=0.
      WT=0.
C     SELECTS MODE OF COMPUTATION
      GO TO(5,10,15),KACE
C
C     WING ALONE
C
    5 CONTINUE
      NI=1
    7 IF(THKW.EQ.0.) GO TO 8
      CALL EVAL1(NI,1.,XB,YB,ZB,UT,VT,WT,NPANEL)                          7
      U=U+UT
      V=V+VT
      W=W+WT
      IF(ISTR.EQ.0) WRITE(N1) XB,YB,ZB,UT,VT,WT                           9
    8 CONTINUE
      IF(CAMN.EQ.1.) GO TO 23
      CALL EVAL1(NI,0.,XB,YB,ZB,UT,VT,WT,NPANEL)                         15
      U=U+UT
      V=V+VT
      W=W+WT
      IF(ISTR.EQ.0) WRITE(N2) XB,YB,ZB,UT,VT,WT                          17
      IF(KACE.NE.3) GO TO 23
      CALL EVAL1(1,0.,XB,YB,ZB,UT,VT,WT,NBODY)             LACKEY        22
      U=U+UT                                               LACKEY
      V=V+VT                                               LACKEY
      W=W+WT                                               LACKEY
      IF(ISTR.EQ.0) WRITE(N3) XB,YB,ZB,UT,VT,WT                          24
   23 CONTINUE
      GO TO (18,18,10),KACE
C     COMPUTES FOR BODY ALONE
   10 CALL VEL1(XB,YB,ZB,UT,VT,WT)                                       29
      U=U+UT
      V=V+VT
      W=W+WT
      IF(ISTR.EQ.0) WRITE(N4) XB,YB,ZB,UT,VT,WT                          31
      GO TO(5,18,18),KACE
C
C     BODY AND WING
   15 NI=NBODY+1
      GO TO 7
   18 CONTINUE
      IF (CPCALC-1.) 320,320,335
  320 CONTINUE
```

```
C          LINEAR CP FORMULA
      CP = -2.*U
      GO TO 340
  330 CONTINUE
C          NON-LINEAR CP FORMULA
      CP = -2.*U+BT2*U*U-V*V-W*W
      GO TO 340
C           **EXACT**  ISENTROPIC CP FORMULA
  335 Q2=(U+COSARA)**2+V*V+(W+SINARA)**2
      CP = 1.42857/XMACH2*((1.+.2*XMACH2*(1.-Q2))**3.5-1.)                 43
  340 CONTINUE
      U = U + COSARA
      W = W + SINARA
   20 RETURN
      END
```

```
   LENG                                                                   07/31/65
 7094 RELMOD ASSEMBLY.

        $IBLDR LENG                                                                         LENG0000
ASSEMBLED TEXT.

        $TEXT  LENG                                                                         LENG0001

                                *LENG        MICHAEL SYNGE          6-9247      ASSEMBLED  27 AUG 64
                                *     TO OBTAIN THE CONTENTS OF THE DECREMENT OF A
                                *     CODE WORD AS A FULL WORD INTEGER
                                *
                                *            CALLING SEQUENCE
                                *
                                *                 N=LENG(CODE)
                                *
                                        ENTRY   LENG
                                *
BINARY CARD (NOT PUNCHED)
       00000  0500 60 4 00003  10000   LENG   CLA*    3,4
       00001  4320 00 0 00004  10001          ANA     =O77777000000
       00002  0771 00 0 00022  10000          ARS     18
       00003  0020 00 4 00001  10000          TRA     1,4
                                *
       00004  077777000000     10000          *LORG
                         00000 01111          END
    CONTROL DICTIONARY

        $CDICT LENG                                                                         LENG0002

BINARY CARD (NOT PUNCHED)
             000005000000              PREFACE      START=0,LENGTH=5,TYPE=7094,CMPLX=5
             000004000005
             432545276060        LENG  DECK         LOC=0,LENGTH=5
             000005000000
             432545276060        LENG  REAL         LOC=0,LENGTH=0
             000000000000

        $DKEND LENG                                                                         LENG0003

NO MESSAGES FOR THIS ASSEMBLY
      SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

 CLASS  SYMBOL   VALUE   REFERENCES

        LENG     00000
 LCTR    BLCTR
 QUAL    UNQS
 LCTR   //
```

```
    LOC                                                                  06/02/67
 7094 RELMOD ASSEMBLY.

      $IBLOR ILOC                                                                      ILOC0000



          ILOC
        ASSEMBLED TEXT.
              $TEXT  ILOC                                                    ILOC0001

                                                 ENTRY   LOC

 BINARY CARD ID. ILOC0002
      00000  0500 00 4 00003  10000     LOC CLA     3,4
      00001  4320 00 0 00003  10001         ANA     =077777                             DEDIS294
      00002  0020 00 4 00001  10000         TRA     1,4                                 DEDIS295
      00003  000000077777     10000         *LORG
                        00000  01111         END                                         DEDIS296

          ILOC
        CONTROL DICTIONARY
              $CDICT ILOC                                                               ILOC0003


 BINARY CARD ID. ILOC0004
           000004000000              PREFACE     START=0,LENGTH=4,TYPE=7094,CMPLX=5
           000004000005
           314346236060      ILOC    DECK        LOC=0,LENGTH=4
           000004000000
           434623606060      LOC     REAL        LOC=0,LENGTH=0
           000000000000

              $DKEND ILOC                                                               ILOC0005


 NO MESSAGES FOR THIS ASSEMBLY
```

```
      SUBROUTINE MDMATE
C     FORMATION OF DRAG MINIMIZATION MATRIX
C     MAXIMUM SIZE MATRIX INVERSION =112
C     MUST BE 2 PLUS MAXIMUM NUMBER OF WING PANELS(110)
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION WW(115,115),XBAR(210),AREA(210)
      MDEMEN=115
      READ (NTAPEC) NBODY,NWING,XMACH,SYM,KACE                      2
      NPANEL=NBODY+NWING
      READ (NTAPEC) (I,XBAR(I),DUMMY1,DUMMY2,DUMMY3,DUMMY4,DUMMY5
     1,AREA(I),DUMMY6,DUMMY7,DUMMY8,I=1,NPANEL)                     6
      REWIND NTAPEC
                                                                    15
      DO 100 J=1,NWING
      READ (NTAPEE) (WW(I,J),I=1,NWING)                             18
  100 CONTINUE
      REWIND NTAPEE
                                                                    25
      DO 200 I=1,NWING
      I=I
      II=I+NBODY
      DO 200 J=1,I
      JJ=J+NBODY
      WW(I,J)=-WW(I,J)*AREA(II)-WW(J,I)*AREA(JJ)
      WW(J,I)=WW(I,J)
  200 CONTINUE
      NN=NWING+1
      NNN=NWING+2
      DO 300 J=1,NWING
      JJ=J+NBODY
      WW(J,NN)=-AREA(JJ)
      WW(NN,J)=-AREA(JJ)
      XA=-XBAR(JJ)*AREA(JJ)
      WW(J,NNN)=XA
      WW(NNN,J)=XA
  300 CONTINUE
      WW(NN,NN)=0.0
      WW(NN,NNN)=0.0
      WW(NNN,NN)=0.0
      WW(NNN,NNN)=0.0
      CALL FSF(1,NTAPEA,IRR)                                        66
      DO 325 J=1,NNN
  325 WRITE (NTAPEA)(WW(I,J),I=1,NNN)                               71
      END FILE NTAPEA
C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL            77
```

```
      CALL SINVRT (WW,MDEMEN,NN,IRR1,IRR2,SCALE,DFT,NDETXP)           78
      IF (IRR1) 350,400,350
 350  CONTINUE
      WRITE (NTAPED,6000) IRR1,IRR2,SCALE                             81
      REWIND NTAPEA                                                   82
      STOP

 400  CONTINUE
      DO 450 J=1,NN
 450  WRITE (NTAPEA)(WW(I,J),I=1,NN)                                  87
      END FILE NTAPEA                                                 93
      REWIND NTAPEA
                                                                      94
      CALL FSF(1,NTAPEA,IRR)                                          95
      DO 475 J=1,NNN
 475  READ (NTAPEA)(WW(I,J),I=1,NNN)                                 100
      CALL FSF(2,NTAPEA,IRR)

C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL AND CM      107
      CALL SINVRT (WW,MDEMEN,NNN,IRR1,IRR2,SCALE,DET,NDETXP)         109
      IF (IRR1) 350,500,350

 500  CONTINUE
      DO 550 J=1,NNN
 550  WRITE (NTAPEA)(WW(I,J),I=1,NNN)                                116
      END FILE NTAPEA                                                122
      REWIND NTAPEA
                                                                     123
6000  FORMAT(1H1,46HERROR IN INVERSION OF DRAG MINIMIZATION MATRIX
     1,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)

      RETURN
      END
```

```
            MEAN                                                        08/09/65
        EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE MEAN (X,Y,N)
      DIMENSION X(1),Y(1)
C     SUBROUTINE TO SMCOTH DATA
 10   MC=1                                                   ,1
 20   M3=N-3                                                 ,2
      DO 90 I=1,M3                                           ,3
      I=I                                                    ,4
      A=(Y(I+2)-Y(I))/(X(I+2)-X(I))                          ,5
      C=(Y(I+3)-Y(I+1))/(X(I+3)-X(I+1))                      ,6
      B=Y(I)-A*X(I)                                          ,7
      D=Y(I+1)-C*X(I+1)                                      ,8
      IF (A-C) 40,30,4C                                      ,9
 30   IF (B-D) 80,90,80                                      ,10
 40   XINT=(B-D)/(C-A)                                       ,11
      IF (XINT-X(I)) 8C,50,50                                ,12
 50   IF (XINT-X(I+2)) 60,60,80                              ,13
 60   IF (XINT-X(I+1)) 80,70,70                              ,14
 70   IF (XINT-X(I+3)) 90,90,80                              ,15
 80   YF=A*X(I+1)+B                                          ,16
      Y(I+1)=(Y(I+1)+YF)/2.                                  ,17
      YS=C*X(I+2)+D                                          ,18
      Y(I+2)=(Y(I+2)+YS)/2.                                  ,19
 90   CONTINUE                                               ,20   ,21
      MC=MC+1                                                ,22
      IF (MC-5) 20,20,100                                    ,23
100   RETURN                                                 ,24
      END                                                    ,25
```

```
          MERR                                                                      07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      FUNCTION MERR(NU,IOK,M,SR)
      DIMENSION NU(3)
C
C     ERROR MESSAGE WRITER FOR GEOMD.
C     NU(1) IS AN ERROR RETURN FROM SOME SUBROUTINE, AND IT SHOULD
C     EQUAL IOK.  IF SO, MERR = 0 AND RETURN.
C     IF NOT, THE ERROR COUNTER, NU(3), IS DECREASED BY 1 AND A
C     MESSAGE IS WRITTEN IF THE TAPE NO., NU(2), .GT. 0 AND
C     NU(3) .GT. 0.  SR IS ALPHAMERIC NAME OF SUBROUTINE.
C
      MERR = 0                                                            ,1
      IF(NU-IOK)100,500,100                                               ,2
  100 LT= NU(2)                                                           ,3
      NU(3)=NU(3)-1                                                       ,4
      IF((LT .LE. 0) .OR. (NU(3) .LE. 0)) GO TO 400                       ,5    ,6    ,7
      WRITE (LT,200) M,NU(1),SR                                           ,8    ,9    ,10
  200 FORMAT( 6HOERROR I6,6H, CODE I6,15H IN SUBROUTINE A6,
     1 28H DURING GEOMETRIC DEFINITION)
      MERR=1                                                              ,11
      GO TO 500                                                           ,12
  400 MERR=-1                                                             ,13
  500 RETURN                                                              ,14
      END                                                                 ,15
```

```
      FUNCTION NWEED(A,N,J,EP)
      DIMENSION A(J,N),EP(1)
C
C     A IS AN ARRAY OF N POINTS IN J SPACE.
C     ELIMINATE ANY POINT WHICH IS COINCIDENT OR NEARLY SO (WITHIN FP)
C        WITH THE PRECEDING POINT.
C     SHIFT REMAINING POINTS TO TOP OF ARRAY.
C     THE LAST POINT IN A AT INPUT WILL BE THE LAST POINT AT OUTPUT.
C     NWEED = NO. OF POINTS IN A AT OUTPUT.
C
      IA=1                                                          ,1
      DO1000 IB=2,N
C                                                                   ,2
      DO 100 IC=1,J                                                 ,3
      IF (ABS(A(IC,IA)-A(IC,IB))-EP(IC))100,300,300                 ,4
 100  CONTINUE
C                                                                   ,5    ,6
 200  IF (N-IB)400,400,1000                                         ,7
 300  IA=IA+1
C                                                                   ,8
 400  DO 500 IC=1,J                                                 ,9
 500  A(IC,IA)=A(IC,IB)
C                                                                   ,10   ,11
 1000 CONTINUE
C                                                                   ,12   ,13
      NWEED=IA                                                      ,14
      RETURN                                                        ,15
      END                                                           ,16
```

```
              OPCAM                                                        08/19/65
          EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C      CONTROL LINK
       DIMENSION DICT(7)
       COMMON DAT(2),LA,LB,LC,LD,LE,LF,LI,LO,
      1 NBODY,NWING,XMACH,SYM,KACE
      2 /LGEOMD/LGDEF(3,6)
       LOGICAL LGDEF
       DATA DICT/6HDEFINE,6HDEFINE,5HPANEL,6HAERODY,6HAERODY,
      1 6HEND OF,6HEND OF/
C
C      INITIALIZE
       CALL OPCAMI
C
C      READ COMMAND CARD
C                                                                         ,1
 100   IGO=INTURP(DICT,7,LI,LO)
C
C      THE FOLLOWING THREE STATEMENTS CAN BE REMOVED IF THE TIME AND DATE
C      OPPOSITE EACH COMMAND ARE NOT WANTED OR IF AN APPROPRIATE
C      SUBROUTINE CLOK IS NOT AVAILABLE.  OR ELSE A DUMMY SUBROUTINE CLOK
C      WHICH RETURNS BLANKS FOR TIM AND DAYT CAN BE USED.                 ,2
       CALL CLOK(TIM,DAYT)                                                ,3
       WRITE(LO,105)TIM,DAYT                                              ,4    ,5    ,6
 105   FORMAT(1H+ 101X,4HTIME 2A7)

       IF (IGO)100,100,150
C                                                                         ,7
 150   GO TO (200,200,300,400,400,500,500  ),IGO
C
C      WHEN A NEW COMMAND IS NEEDED,
C        INCREASE DIMENSION OF DICT BY 1
C        ADD COMMAND TO DATA STATEMENT
C        INCREASE 2ND ARGUEMENT OF INTURP BY 1
C        ADD THE NEW STATEMENT NO. TO THE   GO TO   AT 150
C        ADD THE STATEMENT, FOLLOW WITH   GO TO 100
C
C      A WING AND/OR BODY ARE TO BE DEFINED                               ,8
 200   CALL GEOMD(DAT,LI,LO,LO,LA)                                        ,9
       GO TO 100
C
C      CALL PANEL LINK
C      TRANSFORM BODY TO NEW COORD. SYSTEM, FLATTEN WING, INTERSECT BODY
C      AND WING.  THEN CALL PANEL LINK.                                   ,10
  300  CALL TFLAT(DAT,LI,LO,LA,LB,LD,NBODY,NU)                            ,11
       IF(NU)100,310,100
C      NBODY WAS SET TO -1 BY TFLAT IF THIS IS A BODY-ONLY CASE.
C        THEN WE DO NOT CALL THE PANEL LINK.  NBODY = 0 OTHERWISE.        ,12
  310  IF(NBODY+1)320,100,320
C      IF TFLAT DETECTS AN ERROR, IT READS DATA CARDS UNTIL IT
C        HITS AN   END OF DATA   CARD OR A   DEFINE   CARD.
C        THEN THE INPUT TAPE IS BACKSPACED 1 RECORD.                      ,13
  320  CALL PANEL                                                         ,14
       GO TO 100
C
```

```
                OPCAM                                                     08/19/65
           EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C      CALL AERODYNAMIC LINK                                                    ,15
 400   CALL AERO                                                                ,16
       GO TO 100
C
C      END OF   OR  END OF  CARD WAS READ                                       ,17
 500   STOP                                                                     ,18
       END                                                                      ,19
```

```
        OPCAMI                                                        08/17/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C     INITIALIZATION FOR OPCAM
C
      SUBROUTINE OPCAMI
C
      COMMON DAT(2),LA,LB,LC,LD,LF,LF,LI,LO,
     1 NBODY,NWING,XMACH,SYM,KACE
C
      DIMENSION ICARD(14),ICODE(2)
C
      DATA  ICODE/6HEND OF,6HEND OF/
C
C     ASSIGN TAPE NUMBERS.  TAPES 1-4 AND 7-9 ARE SCRATCH TAPES.
C     LIN IS THE SYSTEM INPUT TAPE.  DATA IS TRANSFERRED FROM LIN TO LI,
C       AS SHOWN BELOW.  LO IS THE SYSTEM OUTPUT TAPE.
      LA=1                                                          ,1
      LB=2                                                          ,2
      LC=3                                                          ,3
      LD=4                                                          ,4
      LIN=5                                                         ,5
      LO=6                                                          ,6
      LE=7                                                          ,7
      LF=8                                                          ,8
      LI=9
C                                                                   ,9
      CALL DATE(DAT,DAT(2))                                         ,10
      WRITE(LO,100)DAT                                              ,11   ,12   ,13
  100 FORMAT(1H1 56X 2A6/69H PROGRAM FOR ANALYSIS AND DESIGN OF SUPERSON
     .IC WING-BODY COMBINATIONS/
     .28HODATA CARDS ARE LISTED BELOW//)
C
C     READ DATA CARDS FROM TAPE  LIN, WRITE THEM ON TAPE  LO
C     (AS PART OF JOB PRINTOUT), AND ALSO WRITE THEM ON TAPE  LI
C     WHICH WILL BE USED AS THE INPUT TAPE FOR REST OF JOB.
      REWIND LI                                                     ,14
      DO 1000 I=1,10000                                             ,15
      READ (LIN,200) ICARD                                          ,16   ,17   ,18
  200 FORMAT(13A6,A2)
      WRITE (LI,200) ICARD                                          ,19   ,20   ,21
      WRITE (LO,300) ICARD                                          ,22   ,23   ,24
  300 FORMAT(1H 14A6)
      DO 400 J=1,2                                                  ,25
      IF(ICARD-ICODE(J)) 400,2000,400                               ,26
  400 CONTINUE                                                      ,27   ,28
 1000 CONTINUE
C                                                                   ,29   ,30
 2000 CONTINUE                                                      ,31
      END FILE LI                                                   ,32
      REWIND LI                                                     ,33
      WRITE(LO,5000) I                                              ,34   ,35   ,36
 5000 FORMAT( 1HO 5X 10HA TOTAL OF I5,22H  DATA CARDS WERE READ/1H1)
C
      RETURN                                                        ,37
      END                                                           ,38
```

```
C                                                                       OPTI0030
      SUBROUTINE OPTIM3(X,Y,ZEE,A,B,K,L,M)                              OPTI0040
C                                                                       OPTI0050
C          INTERPOLATES BETWEEN GIVEN POINTS, CHOOSING AND INSERTING    OPTI0060
C          IN EACH INTERVAL THE NECESSARY AND SUFFICIENT NUMBER OF      OPTI0070
C          INTERMEDIATE POINTS TO MAKE THE MAXIMUM DISTANCE FROM CHORD  OPTI0080
C          TO ARC NO MORE THAN A GIVEN CONSTANT, Z.  THE FUNCTION USED  OPTI0090
C          HAS A CONTINUOUS FIRST DERIVATIVE.                           OPTI0100
C                                                                       OPTI0110
      DIMENSION X(4),Y(4),A(200),B(200),C(3,2),S(3)                     OPTI0120
C                                                                       OPTI0140
      EPS=.002                                                          OPTI0150   ,1
      CALL DVCHK (KOOOFX)                                               OPTI0160   ,2
       GO TO(1,1),KOOOFX                                                OPTI0170   ,3
    1 IF (ZEE) 2,2,3                                                    OPTI0180   ,4
    2 Z=EPS*ABS(X(K+1)-X(K))                                            OPTI0190   ,5
      GO TO 4                                                           OPTI0200   ,6
    3 Z=ZEE                                                             OPTI0210   ,7
    4 KEY=0                                                             OPTI0220   ,8
      MU=0                                                              OPTI0230   ,9
      NU=0                                                              OPTI0240   ,10
      IF (X(1)-X(4)) 12,5,10                                            OPTI0250   ,11
    5 M=0                                                               OPTI0260   ,12
      GO TO 100                                                         OPTI0270   ,13
   10 MU=1                                                              OPTI0280   ,14
      DO 11 I=1,4                                                       OPTI0290   ,15
   11 X(I)=-X(I)                                                        OPTI0300   ,16   ,17
   12 I=1                                                               OPTI0310   ,18
      J=2                                                               OPTI0320   ,19
      IF (K-2) 13,15,14                                                 OPTI0330   ,20
   13 J=1                                                               OPTI0340   ,21
      GO TO 19                                                          OPTI0350   ,22
   14 I=2                                                               OPTI0360   ,23
      GO TO 19                                                          OPTI0370   ,24
   15 DO 18 N=1,3                                                       OPTI0380   ,25
      S(N)=(Y(N+1)-Y(N))/(X(N+1)-X(N))                                  OPTI0390   ,26
      IF (N-2) 18,16,16                                                 OPTI0400   ,27
   16 IF (ABS(S(N)-S(N-1))-EPS) 17,18,18                                OPTI0410   ,28
   17 I=N-1                                                             OPTI0420   ,29
      J=N-1                                                             OPTI0430   ,30
      NU=1                                                              OPTI0440   ,31
      GO TO 19                                                          OPTI0450   ,32
   18 CONTINUE                                                          OPTI0460   ,33   ,34
   19 DO 20 N=I,J                                                       OPTI0470   ,35
      J1=N                                                              OPTI0480   ,36
      J2=N+1                                                            OPTI0490   ,37
      J3=N+2                                                            OPTI0500   ,38
      D=(Y(J2)-Y(J1))/(X(J2)-X(J1))                                     OPTI0510   ,39
      C(3,N)=((Y(J3)-Y(J1))/(X(J3)-X(J1))-D)/(X(J3)-X(J2))              OPTI0520   ,40
      C(2,N)=D-C(3,N)*(X(J2)+X(J1))                                     OPTI0530   ,41
   20 C(1,N)=Y(J1)-X(J1)*C(2,N)-X(J1)**2*C(3,N)                         OPTI0540   ,42   ,43
      DX=X(3)-X(2)                                                      OPTI0550   ,44
      D=0.0                                                             OPTI0560   ,45
```

```
      OPTIM3                                                       09/14/65
   EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      E=X(K)                                                        OPTI0570   ,46
      DO 85 I=1,200                                                 OPTI0580   ,47
      L=I                                                           OPTI0590   ,48
      E=E+D                                                         OPTI0600   ,49
      IF (E+.01*(X(K+1)-X(K))-X(K+1)) 40,35,35                      OPTI0610   ,50
   35 L=L-1                                                         OPTI0620   ,51
      GO TO 90                                                      OPTI0630   ,52
   40 A(L)=E                                                        OPTI0640   ,53
      IF (NU) 45,42,45                                              OPTI0650   ,54
   42 P=AMIN1(AMAX1((X(3)-E)/DX,0.0),1.0)                           OPTI0660   ,55
      GO TO 50                                                      OPTI0670   ,56
   45 P=2-J                                                         OPTI0680   ,57
   50 Q=1.0-P                                                       OPTI0690   ,58
      F=P*(C(1,1)+E*(C(2,1)+E*C(3,1)))+Q*(C(1,2)+E*(C(2,2)+E*C(3,2)))  OPTI0700   ,59
      B(L)=F                                                        OPTI0710   ,60
      G=P*(C(2,1)+2.0*E*C(3,1))+Q*(C(2,2)+2.0*E*C(3,2))             OPTI0720   ,61
      T=ATAN(G)                                                     OPTI0730   ,62
      W=Z*COS(T)                                                    OPTI0740   ,63
      H=2.0*(P*C(3,1)+Q*C(3,2))                                     OPTI0750   ,64
   60 R=(SQRT(1.0+G**2))**3/ABS(H)                                  OPTI0760   ,65
      CALL DVCHK (KOOOFX)                                           OPTI0770   ,66
       GO TO(80,65),KOOOFX                                          OPTI0780   ,67
   65 IF (R-W) 70,70,75                                             OPTI0790   ,68
   70 D=W                                                           OPTI0800   ,69
      GO TO 85                                                      OPTI0810   ,70
   75 V=ACOS(1.0-W/R)                                               OPTI0820   ,71
      U=T+SIGN(V,H)                                                 OPTI0830   ,72
      D=2.8284*COS(U)*SQRT(R*W)                                     OPTI0840   ,73
      GO TO 85                                                      OPTI0850   ,74
   80 IF (KEY) 90,82,9C                                             OPTI0860   ,75
   82 KEY=1                                                         OPTI0870   ,76
      D=.2*(X(K+1)-X(K))                                            OPTI0880   ,77
   85 CONTINUE                                                      OPTI0890   ,78   ,79
      IF (E+D-X(K+1)) 88,90,90                                      OPTI0900   ,80
   88 M=2                                                           OPTI0910   ,81
      GO TO 91                                                                 ,82
   90 M=1                                                           OPTI0930   ,83
   91 B(1)=Y(K)                                                     OPTI0940   ,84
      IF (MU) 92,100,92                                             OPTI0950   ,85
   92 DO 95 I=1,4                                                   OPTI0960   ,86
   95 X(I)=-X(I)                                                    OPTI0970   ,87   ,88
      DO 98 I=1,L                                                   OPTI0980   ,89
   98 A(I)=-A(I)                                                    OPTI0990   ,90   ,91
  100 RETURN                                                        OPTI1000   ,92
      END                                                           OPTI1010   ,93
```

```
               OPTMW                                                      08/16/65
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

       SUBROUTINE OPTMW (NW,NTAPEX,A,B,CONSNT,CLBAR,XCPBAR,RFAREA,AREA,CL
      1)
C      OPTIMIZE WING ONLY CASE
       DIMENSION A(1),B(1),AREA(1),CL(1)
       NN=NW+1                                                    ,1
       B(NN)=CLBAR*RFAREA
C      TEST IF PITCHING MOMENT CONSTRAINT                         ,2
       IF (CONSNT) 100,150,100                                    ,3
 100   CONTINUE                                                   ,4
       NN=NW+2                                                    ,5
       B(NN)=XCPBAR*CLBAR*RFAREA
                                                                  ,6
 150   CONTINUE                                                   ,7
       DO 200 J=1,NW                                              ,8
 200   CL(J)=0.                                                   ,9   ,10
       DO 350 J=1,NN                                              ,11
       READ (NTAPEX) (A(I),I=1,NN)                                ,12  ,13  ,14  ,15  ,16
       IF (J-NW) 350,350,250                                      ,17
 250   CONTINUE                                                   ,18
       DO 300 I=1,NW                                              ,19
 300   CL(I)=CL(I)+A(I)*B(J)                                      ,20  ,21
 350   CONTINUE
                                                                  ,22  ,23
       RETURN                                                     ,24
       END                                                        ,25
```

```
      SUBROUTINE OPTMWB (NW,NB,NTAPEX,NTAPEY,THICK,A,B,ALPHA,ABX,ALPHAX,
     1 AREA,CONSNT,CLBAR,XCPBAR,RFAREA,CL)
C     OPTIMIZE WING-BODY CASE
      DIMENSION A(1),B(1),ALPHA(1),ALPHAX(1),AREA(1),CL(1)
      DIMENSION ABX(1)
      IF (THICK) 50,25,50
 25   CONTINUE
      DO 40 I=1,NB
 40   B(I)=0.
      GO TO 75
 50   CONTINUE
      DO 60 I=1,NB
   60 B(I)=-ABX(I)
 75   CONTINUE
      DO 100 J=1,NW
 100  ALPHA(J)=ALPHAX(J)
      DO 150 J=1,NB
      READ (NTAPEY) (A(I),I=1,NW)                                32
      DO 150 I=1,NW
 150  ALPHA(I)=ALPHA(I)-A(I)*B(J)
      DO 200 J=1,NW
 200  B(J)=AREA(J)*ALPHA(J)
      NN=NW+1
      B(NN)=CLBAR*RFAREA
C     TEST IF PITCHING MOMENT CONSTRAINT
      IF (CONSNT) 275,325,275
 275  CONTINUE
      NN=NW+2
      B(NN)=XCPBAR*CLBAR*RFAREA
 325  CONTINUE
      DO 350 J=1,NW
 350  CL(J)=0.
      DO 400 J=1,NN
      READ (NTAPEX) (A(I),I=1,NN)                                72
      DO 400 I=1,NW
 400  CL(I)=CL(I)+A(I)*B(J)
      RETURN
      END
```

```
      OUTB                                                                08/14/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETA,Z)

C     BODY OUTPUT FORMAT

      DIMENSION Z(1),THETA(1)
      WRITE (NTAPEO,6005) (THETA(I),I=1,NTHETA)                            ,1   ,2   ,3   ,4   ,5
      WRITE (NTAPEO,6010)                                                  ,6   ,7
      DO 100 J=1,NROWB                                                     ,8
  100 WRITE (NTAPEO,6015) (J,(Z(I),I=J,NBODY,NROWB))                       ,9   ,10  ,11  ,12  ,13  ,14
                                                                           ,15
 6005 FORMAT(1HO,4X,11HTHETA(DEG.),F14.4,9F10.4,(F29.4,9F10.4))
 6010 FORMAT(1HO,6X,7HROW NO.)
 6015 FORMAT(1H ,I10,F19.5,9F10.5,(F29.5,9F10.5))
      RETURN                                                               ,16
      END                                                                  ,17
```

```
      SUBROUTINE OUTPTB
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
     1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
     2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
     3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
     4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
     50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
      DIMENSION C(3,6)
      DIMENSION ND3(6108)
      EQUIVALENCE (ND3,XCOR)

C     SUBROUTINE TO OUTPUT BODY PANEL DATA

C     DEFINE CONSTANTS
      NBODY=0                                                        ,1
      KODEB=0                                                        ,2
      KODEI=0                                                         3
      NUMBER=0                                                       ,4
      NPART1=1                                                       ,5
      NPART2=2                                                        6
      NPART3=3                                                       ,7
      E=SYM                                                          ,8
      F=0.00001                                                      ,9
      REWIND NTAPE2

C     DO-LOOP TO SET SMALL NEGATIVE AND POSITIVE
C     VALUES EQUAL TO ZERO                                           ,10
      DO 20 I=1,6108                                                 ,11
      IF (ABS(ND3(I))-F) 10,10,20                                    ,12
   10 ND3(I)=0.                                                      ,13
   20 CONTINUE

C     WRITE PRINTOUT HEADINGS                                        ,14   ,15
      WRITE (NWRITE,900)                                             ,16   ,17
      WRITE (NWRITE,910)

C     WRITE TAPE OUTPUT - NUMBER IS PANEL NUMBER
C     NPART1,NPART2 AND NPART3 INDICATE 1-, 2- AND
C     3-PART PANELS      XCOR,YCOR,ZCOR
C     ARE PANEL CORNER POINT COORDINATES                             ,18   ,19
   50 DO 130 I=1,NPER1                                               ,20
      DO 130 J=1,NPLN1                                               ,21
      KGO=KPANEL(I,J)                                                ,22
      GO TO (52,53,52,53),KGO

C     SETUP OUTPUT ARRAY                                             ,23
   52 C(1,1)=XCOR(I,J+1)                                             ,24
      C(2,1)=YCOR(I,J+1)                                             ,25
```

```
      C(3,1)=ZCOR(I,J+1)                                          ,26
      C(1,2)=XCOR(I,J)                                            ,27
      C(2,2)=YCOR(I,J)                                            ,28
      C(3,2)=ZCOR(I,J)                                            ,29
      C(1,3)=XINT(I,J,1)                                          ,30
      C(2,3)=YINT(I,J,1)                                          ,31
      C(3,3)=ZINT(I,J,1)                                          ,32
      GO TO 55                                                    ,33
 53   C(1,1)=XCOR(I,J)                                            ,34
      C(2,1)=YCOR(I,J)                                             35
      C(3,1)=ZCOR(I,J)                                            ,36
      C(1,2)=XINT(I,J,1)                                          ,37
      C(2,2)=YINT(I,J,1)                                          ,38
      C(3,2)=ZINT(I,J,1)                                           39
      C(2,3)=YCOR(I,J+1)                                          ,40
      C(1,3)=XCOR(I,J+1)                                          ,41
      C(3,3)=ZCOR(I,J+1)                                          ,42
 55   GO TO (57,57,58,58),KGO                                      43
 57   C(1,4)=XCOR(I+1,J)                                          ,44
      C(2,4)=YCOR(I+1,J)                                          ,45
      C(3,4)=ZCOR(I+1,J)                                          ,46
      C(1,5)=XINT(I,J,2)                                          ,47
      C(2,5)=YINT(I,J,2)                                          ,48
      C(3,5)=ZINT(I,J,2)                                          ,49
      C(1,6)=XCOR(I+1,J+1)                                        ,50
      C(2,6)=YCOR(I+1,J+1)                                        ,51
      C(3,6)=ZCOR(I+1,J+1)                                        ,52
      GO TO 60                                                    ,53
 58   C(1,4)=XINT(I,J,2)                                          ,54
      C(2,4)=YINT(I,J,2)                                          ,55
      C(3,4)=ZINT(I,J,2)                                          ,56
      C(1,5)=XCOR(I+1,J+1)                                        ,57
      C(2,5)=YCOR(I+1,J+1)                                        ,58
      C(3,5)=ZCOR(I+1,J+1)                                        ,59
      C(1,6)=XCOR(I+1,J)                                          ,60
      C(2,6)=YCOR(I+1,J)                                          ,61
      C(3,6)=ZCOR(I+1,J)

C     CHECK SUBPANELS AGAINST TOLERANCE                           ,62
 60   KTYPE=1                                                     ,63
      GO TO (63,64,63,64),KGO                                     ,64
 63   JTEMP=J+1                                                   ,65
      GO TO 65                                                    ,66
 64   JTEMP=J                                                      67
 65   SC=SQRT((YCOR(I,JTEMP)-YINT(I,J,1))**2+(ZCOR(I,JTEMP)-      ,68
     1ZINT(I,J,1))**2)/(XCOR(I,J+1)-XCOR(I,J))
      IF (ABS(SC)-E) 67,67,70                                     ,69
 67   KTYPE=2                                                     ,70
 70   GO TO (73,73,74,74),KGO                                     ,71
 73   JTEMP=J+1                                                    72
      GO TO 75                                                    ,73
 74   JTEMP=J                                                     ,74
 75   SC=SQRT((YCOR(I+1,JTEMP)-YINT(I,J,2))**2+(ZCOR(I+1,JTEMP)-  ,75
     1ZINT(I,J,2))**2)/(XCOR(I+1,J+1)-XCOR(I+1,J))
```

```
      OUTPTB                                                        08/16/65
        EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      IF (ABS(SC)-E) 77,77,80                                        76
 77   KTYPE=KTYPE+2                                                 ,77
 80   NUMBER=NUMBER+1                                               ,78
      GO TO (90,100,110,120),KTYPE                                  ,79
 90   WRITE (NWRITE,920) NUMBER,NPART3,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)               ,80   ,81   ,82   ,83   ,84   ,85
                                                                    ,86   ,87   ,88   ,89   ,90   ,91
                                                                    ,92   ,93   ,94

      WRITE (NTAPE2) NUMBER,NPART3,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)               ,95   ,96   ,97   ,98   ,99   ,100
                                                                    ,101  ,102  ,103  ,104  ,105  ,106
                                                                    ,107  ,108  ,109

      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                      ,110  ,111  ,112  ,113  ,114  ,115
                                                                    ,116  ,117  ,118  ,119  ,120  ,121
                                                                    ,122  ,123

      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                      ,124  ,125  ,126  ,127  ,128  ,129
                                                                    ,130  ,131  ,132  ,133  ,134  ,135
                                                                    ,136  ,137

      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                        ,138  ,139  ,140  ,141  ,142  ,143
                                                                    ,144  ,145  ,146  ,147  ,148  ,149
                                                                    ,150  ,151

      WRITE (NTAPE2) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                        ,152  ,153  ,154  ,155  ,156  ,157
                                                                    ,158  ,159  ,160  ,161  ,162  ,163
                                                                    ,164  ,165
      GO TO 130                                                     ,166
100   WRITE (NWRITE,920) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)             ,167  ,168  ,169  ,170  ,171  ,172
                                                                    ,173  ,174  ,175  ,176  ,177  ,178
                                                                     179  ,180  ,181

      WRITE (NTAPE2) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)             ,182  ,183  ,184  ,185  ,186  ,187
                                                                    ,188  ,189  ,190  ,191  ,192  ,193
                                                                    ,194  ,195  ,196

      WRITE (NWRITE,930) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                        ,197  ,198  ,199  ,200  ,201  ,202
                                                                    ,203  ,204  ,205  ,206  ,207  ,208
                                                                    ,209  ,210

      WRITE (NTAPE2) ((C(K,L),K=1,3),L=4,6,2),
     1((C(K,L),K=1,3),L=5,6)                                        ,211  ,212  ,213  ,214  ,215  ,216
                                                                    ,217  ,218  ,219  ,220  ,221  ,222
                                                                     223  ,224
      GO TO 130                                                     ,225
110   WRITE (NWRITE,920) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)               ,226  ,227  ,228  ,229  ,230  ,231
                                                                    ,232  ,233  ,234  ,235  ,236  ,237
                                                                    ,238  ,239  ,240

      WRITE (NTAPE2) NUMBER,NPART2,
     1((C(K,L),K=1,3),L=1,2),((C(K,L),K=1,3),L=1,3,2)               ,241  ,242  ,243  ,244  ,245  ,246
                                                                    ,247  ,248  ,249  ,250  ,251  ,252
                                                                    ,253  ,254  ,255
```

```
      WRITE (NWRITE,930)
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                  ,256  ,257  ,258  ,259  ,260  ,261
                                                                          ,262  ,263  ,264  ,265  ,266  ,267
                                                                          ,268  ,269

      WRITE (NTAPE2)
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                  ,270  ,271  ,272  ,273  ,274  ,275
                                                                          ,276  ,277  ,278  ,279  ,280  ,281
                                                                           282  ,283
      GO TO 130                                                           ,284
  120 WRITE (NWRITE,920) NUMBER,NPART1,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                  ,285  ,286  ,287  ,288  ,289  ,290
                                                                           291  ,292  ,293  ,294  ,295  ,296
                                                                          ,297  ,298  ,299

      WRITE (NTAPE2) NUMBER,NPART1,
     1((C(K,L),K=1,3),L=2,4,2),((C(K,L),K=1,3),L=3,5,2)                  ,300  ,301  ,302  ,303  ,304  ,305
                                                                           306  ,307  ,308  ,309  ,310  ,311
                                                                          ,312  ,313  ,314
  130 CONTINUE                                                            ,315  ,316  ,317
      NBODY=NUMBER                                                        ,318
      NUMBER=0

C     WRITE PRINTOUT HEADINGS                                             ,319
      WRITE (NWRITE,940)                                                  ,320  ,321
      WRITE (NWRITE,950)

C     WRITE TAPE OUTPUT - XCEN,YCEN,ZCEN ARE
C     PANEL CENTROID COORDINATES      XCON,YCON,ZCON
C     ARE PANEL CONTROL POINT COORDINATES      AREA IS
C     PANEL AREA     THETA AND ALPHA ARE PANEL THETA-
C     INCLINATION AND ALPHA-INCIDENCE SLOPES                              ,322  ,323
  150 DO 160 I=1,NPER1                                                    ,324
      DO 160 J=1,NPLN1                                                    ,325
      NUMBER=NUMBER+1                                                     ,326
      WRITE (NWRITE,960) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZCEN(I,J),XCON(I,J),YCON(I,J),ZCON(I,J),AREA(I,J),
     2THETA(I,J),ALPHA(I,J)                                               ,327  ,328  ,329
  160 WRITE (NTAPE2) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZCEN(I,J),XCON(I,J),YCON(I,J),ZCON(I,J),AREA(I,J),
     2THETA(I,J),ALPHA(I,J),CHORD(I,J)

C     WRITE TAPE OUTPUT - NUMBER OF PANELS PER
C     COLUMN AND FRACTION OF PANEL STREAMWISE
C     CHORD FOR PANEL CONTROL LOCATION                                    ,330  ,331  ,332  ,333  ,334
  200 NPAN=NBODY/NPER1                                                    ,335
      WRITE (NTAPE2) NPAN,XPER                                            ,336  ,337  ,338
  250 RETURN                                                               339
  900 FORMAT (1H1,9X,35HBODY PANEL CORNER POINT COORDINATES/10X,
     186H1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
  910 FORMAT (1H0,5X,5HPANEL,    4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
  920 FORMAT (1H ,3X,I3,3X,I1,3X,12F9.5)
  930 FORMAT (1H ,13X,12F9.5)
```

```
      OUTPTB                                                        08/16/65
         EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

940  FORMAT (1H1,9X,49HBODY PANEL CENTROID AND CONTROL POINT COORDINATE
    1S)
950  FORMAT (1H0,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,9X,
    16HTHETA-,3X,6HALPHA-/13X,3(1HC,8X),3(2HCP,7X),17X,6HINCLIN,3X,
    25HINCID//)
960  FORMAT (1H ,1X,I3,4X,6F9.5,5X,F9.5,5X,2F9.5)
      END                                                            ,340
```

```
       SUBROUTINE OUTPTW
       COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
      1NWRITE,NBODY,NWING,XMACH,SYM,KACE
       COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
      1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
      2,NPLANE,NPLN1
       COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
      1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
      2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
      3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
      4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
      5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
       DIMENSION C(3,5)
       DIMENSION ND3(3333)
       EQUIVALENCE (ND3,XCOR)
C      SUBROUTINE TO OUTPUT WING PANEL DATA
C      DEFINE CONSTANTS
       NWING=0                                                   ,1
       KODEW=0                                                   ,2
       NUMBER=0                                                  ,3
       NPART1=1                                                  ,4
       NPART2=2                                                  ,5
       E=SYM                                                     ,6
       F=0.00001                                                 ,7
       ZHT=ZCOR(1,1)
C      DO-LOOP TO SET SMALL NEGATIVE AND POSITIVE
C      VALUES EQUAL TO ZERO                                      ,8
       DO 20 I=1,3333                                            ,9
       IF (ABS(ND3(I))-F) 10,10,20                               ,10
 10    ND3(I)=0.                                                 ,11
 20    CONTINUE
C      WRITE PRINTOUT HEADINGS                                   ,12   ,13
       WRITE (NWRITE,900)                                        ,14   ,15
       WRITE (NWRITE,910)
C      WRITE TAPE OUTPUT - NUMBER IS PANEL NUMBER
C      NPART1,NPART2 INDICATE 1- AND 2-PART PANELS
C      XCOR,YCOR AND ZCOR ARE PANEL CORNER POINT
C      COORDINATES
C      IF KSTART = 2, WRITE OUTPUT DATA FOR
C      INTERSECTION REGION                                       ,16   ,17
       IF (KSTART-1) 150,40,21                                   ,18
 21    DO 39 J=1,NPER1                                           ,19
       I=1                                                       ,20
       NUMBER=NUMBER+1                                           ,21
       KGO=KPANEL(1,J)                                           ,22
       IF (KGO-1) 150,23,24
```

```
          OUTPTW                                                     09/14/65
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C      SETUP OUTPUT ARRAY                                   ,23
 23    C(1,1)=XCOR(I,J+1)                                   ,24
       C(2,1)=YCOR(I,J+1)                                   ,25
       C(3,1)=ZCOR(I,J+1)                                   ,26
       C(1,2)=XCOR(I,J)                                     ,27
       C(2,2)=YCOR(I,J)                                     ,28
       C(3,2)=ZCOR(I,J)                                     ,29
       C(1,3)=XINT(1,J)                                     ,30
       C(2,3)=YINT(1,J)                                     ,31
       C(3,3)=ZINT(1,J)                                     ,32
       GO TO 25                                             ,33
 24    C(1,1)=XCOR(I,J)                                     ,34
       C(2,1)=YCOR(I,J)                                     ,35
       C(3,1)=ZCOR(I,J)                                     ,36
       C(1,2)=XINT(1,J)                                     ,37
       C(2,2)=YINT(1,J)                                     ,38
       C(3,2)=ZINT(1,J)                                     ,39
       C(1,3)=XCOR(I,J+1)                                   ,40
       C(2,3)=YCOR(I,J+1)                                   ,41
       C(3,3)=ZCOR(I,J+1)                                   ,42
 25    C(1,4)=XCOR(I+1,J)                                   ,43
       C(2,4)=YCOR(I+1,J)                                   ,44
       C(3,4)=ZCOR(I+1,J)                                   ,45
       C(1,5)=XCOR(I+1,J+1)                                 ,46
       C(2,5)=YCOR(I+1,J+1)                                 ,47
       C(3,5)=ZCOR(I+1,J+1)                                 ,48
       KTYPE=1                                              ,49
       GO TO (27,28),KGO

C      CHECK SUBPANELS AGAINST TOLERANCE                    ,50
 27    JTEMP=J+1                                            ,51
       KTEMP=J                                              ,52
       GO TO 30                                             ,53
 28    JTEMP=J                                              ,54
       KTEMP=J+1                                            ,55
 30    SC=(YCOR(I,JTEMP)-YINT(1,J))/(XCOR(I,J+1)-
      1XCOR(I,J))                                           ,56
       IF (ABS(SC)-E) 31,31,32                              ,57
   31 KTYPE=2                                               ,58
   32 IF (YCOR(I+1,KTEMP)-YCOR(I,KTEMP)) 33,33,34           ,59
   33 KTYPE=KTYPE+2                                         ,60
   34 GO TO (36,37,38,140),KTYPE                            ,61
   36 WRITE (NWRITE,920) NUMBER,NPART2,((C(K,L),K=1,3),
      1L=1,2),((C(K,L),K=1,3),L=1,3,2)                      ,62   ,63   ,64   ,65   ,66   ,67
                                                            ,68   ,69   ,70   ,71   ,72   ,73
                                                            ,74   ,75   ,76

       WRITE (NTAPE2) NUMBER,NPART2,((C(K,L),K=1,3),
      1L=1,2),((C(K,L),K=1,3),L=1,3,2)                      ,77   ,78   ,79   ,80   ,81   ,82
                                                            ,83   ,84   ,85   ,86   ,87   ,88
                                                            ,89   ,90   ,91

       WRITE (NWRITE,930) ((C(K,L),K=1,3),L=2,4,2),
      1((C(K,L),K=1,3),L=3,5,2)                             ,92   ,93   ,94   ,95   ,96   ,97
                                                            ,98   ,99   ,100  ,101  ,102  ,103
                                                            ,104  ,105
```

```
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                 ,106  ,107  ,108  ,109  ,110  ,111
                                                               ,112  ,113  ,114  ,115  ,116  ,117
                                                               ,118  ,119
      GO TO 39                                                 ,120
   37 WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                 ,121  ,122  ,123  ,124  ,125  ,126
                                                               ,127  ,128  ,129  ,130  ,131  ,132
                                                               ,133  ,134  ,135
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),L=2,4,2),
     1((C(K,L),K=1,3),L=3,5,2)                                 ,136  ,137  ,138  ,139  ,140  ,141
                                                               ,142  ,143  ,144  ,145  ,146  ,147
                                                               ,148  ,149  ,150
      GO TO 39                                                 ,151
   38 WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                          ,152  ,153  ,154  ,155  ,156  ,157
                                                               ,158  ,159  ,160  ,161  ,162  ,163
                                                               ,164  ,165  ,166
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=1,2),((C(K,L),K=1,3),L=1,3,2)                          ,167  ,168  ,169  ,170  ,171  ,172
                                                               ,173  ,174  ,175  ,176  ,177  ,178
                                                               ,179  ,180  ,181
   39 CONTINUE
C     WRITE OUTPUT TAPE FOR MAIN WING REGION                   ,182  ,183
   40 DO 50 I=KSTART,KEND                                      ,184
      DO 50 J=1,NPER1                                          ,185
      NUMBER=NUMBER+1                                          ,186
      WRITE (NWRITE,920) NUMBER,NPART1,XCOR(I,J),YCOR(I,J),
     1ZCOR(I,J),XCOR(I+1,J),YCOR(I+1,J),ZCOR(I+1,J),XCOR(I,J+1),
     2YCOR(I,J+1),ZCOR(I,J+1),XCOR(I+1,J+1),YCOR(I+1,J+1),ZCOR(I+1,J+1)   ,187  ,188  ,189
   50 WRITE (NTAPE2) NUMBER,NPART1,XCOR(I,J),YCOR(I,J),
     1ZCOR(I,J),XCOR(I+1,J),YCOR(I+1,J),ZCOR(I+1,J),XCOR(I,J+1),
     2YCOR(I,J+1),ZCOR(I,J+1),XCOR(I+1,J+1),YCOR(I+1,J+1),ZCOR(I+1,J+1)
C     IF KEND LESS THAN NPLN1 WRITE OUTPUT DATA
C     FOR WING-TIP REGION     OTHERWISE SKIP THIS
C     SET OF INSTRUCTIONS                                      ,190  ,191  ,192  ,193  ,194
      IF (KEND-NPLN1) 60,85,150                                ,195
   60 DO 80 J=1,NPER1                                          ,196
      NUMBER=NUMBER+1                                          ,197
      KGO=KPANEL(2,J)                                          ,198
      I=NPLN1                                                  ,199
      GO TO (63,64),KGO
C     SETUP OUTPUT ARRAY                                       ,200
   63 C(1,2)=XINT(2,J)                                         ,201
      C(2,2)=YINT(2,J)                                         ,202
      C(3,2)=ZINT(2,J)                                         ,203
      C(1,4)=XCOR(I+1,J+1)                                     ,204
      C(2,4)=YCOR(I+1,J+1)                                     ,205
      C(3,4)=ZCOR(I+1,J+1)                                     ,206
      C(1,5)=XCOR(I+1,J)                                       ,207
      C(2,5)=YCOR(I+1,J)                                       ,208
```

```
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)
                                                                   ,317  ,318  ,319
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=2,5,3),((C(K,L),K=1,3),L=4,5)                              ,320  ,321  ,322  ,323  ,324  ,325
                                                                   ,326  ,327  ,328  ,329  ,330  ,331
                                                                   ,332  ,333  ,334
 80   CONTINUE                                                     ,335  ,336
 85   NWING=NUMBER                                                 ,337
      NUMBER=0
C     WRITE PRINTOUT HEADINGS                                      ,338
      WRITE (NWRITE,940)                                           ,339  ,340
      WRITE (NWRITE,950)
C     WRITE TAPE OUTPUT - XCEN AND YCEN ARE
C     PANEL CENTROID COORDINATES     XCON AND YCON
C     ARE PANEL CONTROL POINT COORDINATES     AREA
C     IS PANEL AREA     ZCEN AND ZCON ARE PANEL
C     THICKNESS AND CAMBER ORDINATES     THETA AND
C     ALPHA ARE PANEL THICKNESS AND CAMBER SLOPES
C     WRITE OUTPUT TAPE                                            ,341  ,342
 90   DO 100 I=1,NPLN1                                             ,343
      DO 100 J=1,NPER1                                             ,344
      NUMBER=NUMBER+1                                              ,345
      WRITE (NWRITE,960) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZHT,XCON(I,J),YCON(I,J),ZHT,AREA(I,J),ZCEN(I,J),THETA(I,J),
     2ZCON(I,J),ALPHA(I,J)                                         ,346  ,347  ,348
 100  WRITE (NTAPE2) NUMBER,XCEN(I,J),YCEN(I,J),
     1ZHT,XCON(I,J),YCCN(I,J),ZHT,AREA(I,J),THETA(I,J),ALPHA(I,J),
     2CHORD(I,J)
C     WRITE TAPE OUTPUT - NUMBER OF PANELS PER
C     COLUMN AND FRACTION OF PANEL STREAMWISE
C     CHORD FOR PANEL CONTROL LOCATION                             ,349  ,350  ,351  ,352  ,353
 110  WRITE (NTAPE2) NPER1,XPER                                    ,354  ,355  ,356
      GO TO 150                                                    ,357
 140  WRITE (NWRITE,970)                                           ,358  ,359
 150  RETURN                                                       ,360
 900  FORMAT (1H1, 9X,35HWING PANEL CORNER POINT COORDINATES/10X,
     186H1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
 910  FORMAT (1HO,5X,5HPANEL,    4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
 920  FORMAT (1H ,3X,I3,3X,I1,3X,12F9.5)
 930  FORMAT (1H ,13X,12F9.5)
 940  FORMAT (1H1, 9X,49HWING PANEL CENTROID AND CONTROL POINT COORDINAT
     1ES)
 950  FORMAT (1HO,1X,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,7X,
     11HZ,9X,6HALPHA-,2X,1HZ,10X,6HALPHA-/14X,3(1HC,8X),3(2HCP,7X),
     216X,2(5HTHICK,4X),2(6HCAMBER,4X)//)
 960  FORMAT (1H ,1X,I3,4X,6F9.5,5X,F9.5,5X,4F9.5)
 970  FORMAT (1H1,33HERROR MESSAGE - SUBROUTINE OUTPTB/10X,54HINBOARD (O
     1R OUTBOARD) PANEL COLUMN HAS ZERO-AREA PANEL/10X,45HNON-CONSTANT N
```

```
        OUTPTW                                                        09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      C(3,5)=ZCOR(I+1,J)                                     ,209
      GO TO 65                                               ,210
 64   C(1,2)=XCOR(I+1,J)                                     ,211
      C(2,2)=YCOR(I+1,J)                                     ,212
      C(3,2)=ZCOR(I+1,J)                                     ,213
      C(1,4)=XINT(2,J)                                       ,214
      C(2,4)=YINT(2,J)                                       ,215
      C(3,4)=ZINT(2,J)                                       ,216
      C(1,5)=XCOR(I+1,J+1)                                   ,217
      C(2,5)=YCOR(I+1,J+1)                                   ,218
      C(3,5)=ZCOR(I+1,J+1)                                   ,219
 65   C(1,1)=XCOR(I,J)                                       ,220
      C(2,1)=YCOR(I,J)                                       ,221
      C(3,1)=ZCOR(I,J)                                       ,222
      C(1,3)=XCOR(I,J+1)                                     ,223
      C(2,3)=YCOR(I,J+1)                                     ,224
      C(3,3)=ZCOR(I,J+1)                                     ,225
      KTYPE=1                                                ,226
      GO TO (67,68),KGC                                      ,227
C     CHECK SUBPANELS AGAINST TOLERANCE
 67   JTEMP=J                                                ,228
      KTEMP=J+1                                              ,229
      GO TO 70                                               ,230
 68   JTEMP=J+1                                              ,231
      KTEMP=J                                                ,232
 70   SC=(YCOR(I+1,JTEMP)-YINT(2,J))/(XCOR(I+1,J+1)-
     1XCOR(I+1,J))                                           ,233
      IF (ABS(SC)-E) 73,73,74                                ,234
 73   KTYPE=2                                                ,235
 74   IF (YCOR(I+1,KTEMP)-YCOR(I,KTEMP)) 75,75,76            ,236
 75   KTYPE=KTYPE+2                                          ,237
 76   GO TO (77,78,79,140),KTYPE                             ,238
 77   WRITE (NWRITE,920) NUMBER,NPART2,((C(K,L),K=1,3),L=1,4)
                                                             ,239  ,240  ,241  ,242  ,243  ,244
                                                             ,245  ,246  ,247
      WRITE (NTAPE2) NUMBER,NPART2,((C(K,L),K=1,3),L=1,4)
                                                             ,248  ,249  ,250  ,251  ,252  ,253
                                                             ,254  ,255  ,256
      WRITE (NWRITE,93C) ((C(K,L),K=1,3),L=2,5,3),
     1((C(K,L),K=1,3),L=4,5)                                 ,257  ,258  ,259  ,260  ,261  ,262
                                                             ,263  ,264  ,265  ,266  ,267  ,268
                                                             ,269  ,270
      WRITE (NTAPE2) ((C(K,L),K=1,3),L=2,5,3),
     1((C(K,L),K=1,3),L=4,5)                                 ,271  ,272  ,273  ,274  ,275  ,276
                                                             ,277  ,278  ,279  ,280  ,281  ,282
                                                             ,283  ,284
      GO TO 80                                               ,285
 78   WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),L=1,4)
                                                             ,286  ,287  ,288  ,289  ,290  ,291
                                                             ,292  ,293  ,294
      WRITE (NTAPE2) NUMBER,NPART1,((C(K,L),K=1,3),L=1,4)
                                                             ,295  ,296  ,297  ,298  ,299  ,300
                                                             ,301  ,302  ,303
      GO TO 80                                               ,304
 79   WRITE (NWRITE,920) NUMBER,NPART1,((C(K,L),K=1,3),
     1L=2,5,3),((C(K,L),K=1,3),L=4,5)                        ,305  ,306  ,307  ,308  ,309  ,310
                                                             ,311  ,312  ,313  ,314  ,315  ,316
```

```
       OUTPTW                                                    09/14/65
   EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

2UMBER OF PANELS FOR THAT COLUMN)
 END                                                                   ,361
```

```
      OUTW                                                             08/14/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE OUTW (INTAPEO,NWING,NCOLW,NROWW,Z)

C     WING OUPUT FORMAT

      DIMENSION Z(1)                                                   ,1    ,2    ,3    ,4    ,5
      WRITE (NTAPEO,6005) (I,I=1,NCOLW)                                ,6    ,7
      WRITE (NTAPEO,6010)                                              ,8
      DO 100 J=1,NROWW                                                 ,9    ,10   ,11   ,12   ,13   ,14
  100 WRITE (NTAPEO,6015) (J,(Z(I),I=J,NWING,NROWW))                   ,15

 6005 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10,(18X,I8,9I10))
 6010 FORMAT(1H ,2X,17HCHORDWISE STATION)
 6015 FORMAT(1H ,I10,F19.5,9F10.5,(F29.5,9F10.5))
      RETURN                                                           ,16
      END                                                              ,17
```

```
       PANEL                                                         08/16/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE PANEL
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/ ND3(11000)
      DIMENSION TABLE(3)

C     CONTROL PROGRAM FOR PANELING SUBROUTINES
      DATA TABLE /10HBODY PANEL,10HWING PANEL,6HPANEND/

C     DEFINE TAPE UNITS AND CONSTANTS
C     ND1, ND4 ARE INPUT SCRATCH TAPES      NTAPE2 IS
C     OUTPUT SCRATCH TAPE
      ND4=ND2(1)                                                    ,1
      NBODY=0                                                       ,2
      NWING=0                                                       ,3
      KINT=0                                                        ,4
      REWIND ND1                                                    ,5
      REWIND ND4                                                    ,6
      REWIND NTAPE2

C     READ CARD INPUT - X- AND Y-PERCENT VALUES FOR
C     BODY AND WING PANEL CONTROL POINT LOCATION                    ,7
      READ (NREAD,890) XPER,YPER

C     DEFINE KODEC      IF KODEC = 0, Y-PRIME FOR
C     CONTROL POINT TO BE SAME AS THAT FOR CENTROID                 ,8    ,9    ,10
      KODEC=1                                                       ,11
      IF (YPER) 150,5,10                                            ,12
    5 KODEC=0                                                       ,13
   10 N=3

C     READ CARD INPUT FOR CONTROL CARD                              ,14
   15 KCODE=INTURP(TABLE,N,NREAD,NWRITE)                            ,15
      IF (KCODE) 15,15,20

C     CALL SUBROUTINES                                              ,16
   20 GO TO (50,100,140),KCODE                                      ,17
   50 CALL BODY                                                     ,18
      IF (KODEB) 150,55,150                                         ,19
   55 IF (KODEI) 150,15,150                                         ,20
  100 CALL WING                                                     ,21
      IF (KODEW) 150,15,150                                         ,22
  140 WRITE (NWRITE,930)                                            ,23   ,24
  150 REWIND NTAPE2                                                 ,25
      RETURN                                                        ,26
  890 FORMAT (6F10.0)
  930 FORMAT (1H1)
       END                                                          ,27
```

```
      SUBROUTINE PARTV
C     PARTITION VELOCITY COMPONENTS MATRIX
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION U(110),V(110),W(110),UU(110),VV(110),WW(110)

      CALL FSF(1,NTAPEA,IRR)                                              2
      CALL FSF(1,NTAPEB,IRR)                                              4
      DO 100 J=1,NBODY
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,
     1NWING)                                                              8
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)                          21
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                       28
  100 CONTINUE
      END FILE NTAPED                                                    37
      REWIND NTAPEA
                                                                         38
      CALL FSF(1,NTAPEA,IRR)                                             39
      DO 200 J=1,NBODY
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                        43
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)                       50
  200 CONTINUE
      END FILE NTAPED                                                    59
      REWIND NTAPEA
                                                                         60
      CALL FSF(1,NTAPEA,IRR)                                             61
      DO 300 J=1,NWING
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,NW
     1ING)                                                               65
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)                          78
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                       85
  300 CONTINUE
      REWIND NTAPEA                                                      94
      REWIND NTAPEB                                                      95
      END FILE NTAPED
                                                                         96
      CALL FSF(1,NTAPEA,IRR)                                             97
      DO 400 J=1,NWING
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)                       101
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)                      108
  400 CONTINUE
      END FILE NTAPED                                                   117
      REWIND NTAPEA                                                     118
      REWIND NTAPED
                                                                        119
      RETURN
      END
```

```
      POLXN                                                         07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE POLXN(C,P,M,N,EP,MAX,Q,LQ,L)
      DIMENSION C(1),P(M,N),Q(M,MAX),LQ(2,1)
C
C     GIVEN A POLYGONAL ARC, P, OF N POINTS IN M-SPACE, FIND ITS
C     INTERSECTIONS, Q, WITH PLANE C.
C
C     L = NO. OF INTERSECTIONS.
C     LQ CONTAINS POINT NUMBERS OF POINTS ADJACENT TO EACH Q.
C
      L=0                                                           ,1
      T=0.                                                          ,2
      DO 20 I=1,M                                                   ,3
 20   T=T+C(I)**2
C                                                                   ,4    ,5
      IF ( T .LE.0. .OR. M .LE. 0 .OR. MAX .LE. 0) GO TO 2000
C     SCALE EP FOR LATER USE                                        ,6    ,7    ,8
 100  E=ABS(EP)*SQRT(T)
C                                                                   ,9
      DO 1000 K=1,N
C                                                                   ,10
      FK=C(M+1)                                                     ,11
      DO 150 I=1,M                                                  ,12
 150  FK=FK+C(I)*P(I,K)
C                                                                   ,13   ,14
      IF (ABS(FK)-E)500,500,600
C                                                                   ,15
 500  U=0.                                                          ,16
      V=-1.                                                         ,17
      K1=K                                                          ,18
      GO TO 900
C                                                                   ,19
 600  IF (K-1)1000,1000,700
C                                                                   ,20
 700  IF (FK*FP)800,1000,1000
C                                                                   ,21
 800  T=FK-FP                                                       ,22
      U=FK/T                                                        ,23
      V=FP/T                                                        ,24
      K1=K-1
C                                                                   ,25
 900  L=L+1                                                         ,26
      DO 920 I=1,M                                                  ,27
 920  Q(I,L)=U*P(I,K-1)-V*P(I,K)
C                                                                   ,28   ,29
      LQ(1,L)=K1                                                    ,30
      LQ(2,L)=K
C                                                                   ,31
      IF (L-MAX)1000,9000,9000
C                                                                   ,32
 1000 FP=FK
C                                                                   ,33   ,34
      GO TO 9000
C                                                                   ,35
```

```
 2000 L=-1
C                                                            ,36
 9000 RETURN                                                 ,37
      END                                                    ,38
```

```
           QRAT                                                          07/31/65
        EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE QRAT(X,Y,IXY,KCDE,XI,YI,C,NU)
      DIMENSION X(1),Y(1),C(3)
C
C     FIND COEFFICIENTS OF QUADRATIC THRU 3 POINTS.
C     1ST PT IS (X,Y), 2ND PT IS (X(IXY+1),Y(IXY+1)),
C     3RD PT IS(X(2*IXY+1),Y(2*IXY+1))
C     IF KODE .GT. 0, EVALUATE AT X=XI
C
C     USING 32769 INSTEAD OF 1, ALLOWS IXY TO BE NEGATIVE.
      JXY=IXY+32769                                                      ,1
      KXY=IXY+JXY                                                        ,2
      DX21=X(JXY)-X                                                      ,3
      DY21=Y(JXY)-Y                                                      ,4
      DX32=X(KXY)-X(JXY)                                                 ,5
      DY32=Y(KXY)-Y(JXY)                                                 ,6
      DX31=X(KXY)-X                                                      ,7
      DY31=Y(KXY)-Y                                                      ,8
      SX31=X(KXY)+X
C                                                                        ,9
 100  IF (DX21*DX32*DX31)300,200,300                                     ,10
 200  NU=1                                                               ,11
      GO TO 1000
C                                                                        ,12
 300  C(3)=(DY32/DX32-DY21/DX21)/DX31                                    ,13
      C(2)=DY31/DX31-C(3)*SX31                                           ,14
      C(1)=Y(JXY)-(C(2)+C(3)*X(JXY))*X(JXY)                              ,15
      NU=0
C                                                                        ,16
 400  IF (KODE)1000,1000,500                                             ,17
 500  YI=C(1)+(C(2)+C(3)*XI)*XI
                                                                         ,18
 1000 RETURN                                                             ,19
      END                                                                ,20
```

```
       READ        -  EFN   SOURCE STATEMENT  -  IFN(S)  -             05/22/67

      SUBROUTINE READ (NTAPEI,NTAPEO,NTAPEC,NM,S)
      DIMENSION S(1),DICT(2)

C     CONTROLS DATA READ IN OPTION FOR SPECIFYING WING CAMBER AND
C     PRESSURE DISTRIBUTION, ALSO BODY CAMBER

      DATA DICT/6HCONSTA,5HGIVEN/

      IGO=INTURP(DICT,2,NTAPEI,NTAPEO)                                    2
      IGO=IGO+1
      GO TO (50,60,70),IGO
 50   CONTINUE
      READ (NTAPEI,5000) (S(I),I=1,NM)                                    5
      GO TO 999
 60   CONTINUE
      READ (NTAPEI,5000) SCONST                                          13
      DO 65 I=1,NM
 65   S(I)=SCONST
      GO TO 999
 70   CONTINUE
      READ (NTAPEC) (S(I),I=1,NM)

 5000 FORMAT(7F10.0)                                                     22

 999  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE REDUCE
C     REDUCE AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND FORM THE
C     (D) AND (E) MATRICES
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF
     1,NTAPEI,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION AB(100),ABB(100,100),ABW(100,25),AWB(110,100),AWW(110,25)
     1),E(100,25),D(110),NSIZE(5)
      EQUIVALENCE (ABB,AWB)
C     COMPUTE SIZE OF PARTIONS
      MAXP=25
      CALL SIZE (NWING,MAXP,NPART,NSIZE)
                                                                      3
      DO 100 J=1,NBODY
  100 READ (NTAPED) (AWB(I,J),I=1,NWING)                             8
      CALL FSF(1,NTAPED,IRR)
C     FORM (D) MATRIX                                                15
      DO 300 K=1,NBODY
      READ (NTAPED) (AB(I),I=1,NBODY)                                19
      DO 250 I=1,NWING
      D(I)=0.0
      DO 250 J=1,NBODY
      D(I)=D(I)+AWB(I,J)*AB(J)
  250 CONTINUE
      WRITE (NTAPEF) (D(I),I=1,NWING)                                35
  300 CONTINUE
      END FILE NTAPEF
                                                                     42
      REWIND NTAPED                                                  43
      CALL FSF(1,NTAPED,IRR)                                         44
      DO 325 J=1,NBODY
  325 READ (NTAPED) (ABB(I,J),I=1,NBODY)
C     COPY INVERSE BODY MATRIX                                       49
      DO 350 J=1,NBODY
  350 WRITE (NTAPEF) (ABB(I,J),I=1,NBODY)
C     MATRIX REDUCTION                                               59
      DO 600 L=1,NPART
      NS=NSIZE(L)
      IF (L .EQ. 1) GO TO 370
      DO 360 J=1,NBODY
  360 READ (NTAPED) (ABB(I,J),I=1,NBODY)                             77
  370 REWIND NTAPED                                                  83
      DO 380 K=1,NS
  380 READ (NTAPEA)(ABW(I,K),I=1,NBODY),(AWW(I,K),I=1,NWING)
C     FORM (E) MATRIX                                                86
      DO 400 K=1,NS
      DO 400 M=1,NBODY
```

```
      E(M,K)=0.
      DO 400 N=1,NBODY
      E(M,K)=E(M,K)-ABB(M,N)*ABW(N,K)
  400 CONTINUE

      DO 450 J=1,NBODY
  450 READ (NTAPED)(AWB(I,J),I=1,NWING)                                113
      CALL FSF(1,NTAPED,IRR)                                           119
      DO 500 K=1,NS
      DO 500 I=1,NWING
      DO 500 M=1,NBODY
      AWW(I,K)=AWW(I,K)+AWB(I,M)*E(M,K)
  500 CONTINUE

      DO 550 K=1,NS
  550 WRITE (NTAPEF)(E(I,K),I=1,NBODY)                                 137
      DO 560 K=1,NS
  560 WRITE (NTAPEE)(AWW(I,K),I=1,NWING)                               145
  600 CONTINUE

      REWIND NTAPEA                                                    153
      REWIND NTAPED                                                    154
      REWIND NTAPEE                                                    155
      END FILE NTAPEF                                                  156
      REWIND NTAPEF
                                                                       157
      RETURN
      END
```

```
       RICH3A                                                          08/16/65
    EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE RICH3A(NP,P,Q,KD,MAX,CHD,NA,A,NU)
      DIMENSION P(3),Q(3),A(3),NU(3),T(12),U(3),V(3)
C
C     P = ARRAY OF NP POINTS, STORED (X1,Y1,Z1,X2,Y2,Z2,.....)
C     Q = SCRATCH ARRAY OF LENGTH 3*NP.  IT MAY BE THE SAME ARRAY
C         AS P IF OK TO DESTROY P.
C
C     RICH3A PERFORMS A 3D INTERPOLATION TO PRODUCE ARRAY A
C     WHICH CONTAINS P WITH INTERSPERSED POINTS.
C
C     NA = NO OF POINTS IN A.
C     MAX = DIMENSION OF A.
C     KD = 0 IF ENRICHING IS DONE IN PLANE THRU 1ST PT, LAST PT,
C       AND PT FARTHEST FROM LINE THRU END PTS.
C     KD = 1 OR 2 IF IN PLANE THRU END PTS WHICH IS NORMAL TO
C       XY PLANE
C     KD = 1 IF PROJECTION OF A ON XY PLANE IS TO BE STRAIGHT LINES
C       BETWEEN POINTS OF P (WHEN PROJECTED TO XY PLANE).
C     KD = 2 OTHERWISE.
C     CHD = CHORD HEIGHT TOLERANCE (SEE ENRYCH)
C
C     INPUTS ARE NP,P,KD,MAX,CHD,NU(2),NU(3)
C     OUTPUTS ARE Q,NA,A,NU(1),NU(3)
C
      MU=0                                                        ,1
      NA=0                                                        ,2
      NP2=NP+NP                                                   ,3
      NP3=NP2+NP                                                  ,4
      MAXR=MAX-NP2
C                                                                 ,5
      IF(NP .LE. 3  .OR.  CHD .LE. 0.) GO TO 7000                 ,6     ,7     ,9
 20   IF(KD)30,30,90
C     FIND ROTATION MATRIX TO SYSTEM WHICH HAS ORIGIN AT 1ST PT,
C       +X AXIS PASSING THRU LAST PT, Z AXIS IN PLANE DESCRIBED FOR KD=0
C
C     FIND THE POINT STARTING IN P(3*JP-2) WHICH IS FARTHEST FROM
C       THE LINE THRU END PTS, AND ITS DISTANCE = DIST.           ,9
 30   CALL DMAXL(P,NP,DIST,JP,U,NU)
C     U IS UNIT VECTOR FROM 1ST PT TO LAST PT.                    ,10
      IF(NU)9000,40,9000                                          ,11
 40   IF(DIST-CHD)7000,7000,50                                    ,12
 50   D=UVECN(P,P(3*JP-2),V,0,3)                                  ,13
      I=1                                                         ,14
      NU=21                                                       ,15
      CALL VCROS(V,U,T(4),D,I)                                    ,16
      IF(I)9000,60,9000
C     T(4)=V CROSS U IS UNIT VECTOR ALONG NEW Y AXIS              ,17
 60   I=1                                                          18
      CALL VCROS(U,T(4),T(7),D,I)                                 ,19
      CALL TRAV(T,U,3)
C     ROTATION MATRIX NOW FORMED.                                 ,20
      T(12)=P(3)                                                  ,21
      GO TO 130
```

```
      RICH3A                                                                08/16/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C
C         KD=1 OR 2
C         T = TRANSFORMATION MATRIX (SEE TROTPT).                                  ,22
   90     DO 100 I=1,12                                                            ,23
  100     T(I)=0.
C                                                                                  ,24    ,25
          T(1)=P(NP3-2)-P(1)                                                       ,26
          T(2)=P(NP3-1)-P(2)                                                       ,27
          D=SQRT(T**2+T(2)**2)                                                     ,28
          IF (D)110,110,120                                                        ,29
  110     NU=7                                                                     ,30
          GO TO 9000
C                                                                                  ,31
  120     T(1)=T(1)/D                                                              ,32
          T(2)=T(2)/D                                                              ,33
          T(9)=1.                                                                   34
          T(4)=-T(2)                                                               ,35
          T(5)=T(1)
C                                                                                  ,36
  130     T(10)=P(1)                                                               ,37
          T(11)=P(2)
C         TRANSLATION MATRIX, IN T(10),T(11),T(12),  NOW FORMED
C
C         TRANSFORM POINTS IN P TO Q                                               ,38
          JX=-2                                                                    ,39
          DO 200 I=1,NP                                                             40
          JX=JX+3                                                                  ,41
  200     CALL TROTPT(T,T(10),2,P(JX),Q(JX))
C                                                                                  ,42    ,43
          IF (MAXR)400,400,500                                                     ,44
  400     NU=1                                                                      45
          GO TO 9000
C                                                                                  ,46
  500     IA=MAXR                                                                   47
          IB=IA+NP                                                                 ,48
          IXQ=IA+1                                                                  49
          IYQ=IB+1                                                                 ,50
          DO 600 I1=3,NP3,3                                                        ,51
          IA=IA+1                                                                   52
          IB=IB+1                                                                  ,53
          A(IA)=Q(I1-2)                                                             54
  600     A(IB)=Q(I1)
C                                                                                  ,55    ,56
          NO=MAXR                                                                  ,57
  700     CALL ENRYCH(A(IXQ),A(IYQ),NP,CHD,A,NO,NU)                                ,58
          IF (NU)800,1000,800                                                      ,59
  800     IF (NU-3)9000,900,9000                                                   ,60
  900     MU=-1                                                                    ,61
 1000     NA=NO/2                                                                  ,62
          IF (MAX-NO-NA)1100,1200,1200                                             ,63
 1100     NU=6                                                                     ,64
          GO TO 9000
C                                                                                  ,65
 1200     IA=NO                                                                     66
```

```
                RICH3A                                                        08/16/65
           EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      DO 1300 I1=1,NO                                                        ,67
      IB=IA+IA/2                                                             ,68
      A(IB)=A(IA)                                                             69
 1300 IA=IA-1
C                                                                            ,70    ,71
      IA=-2                                                                   72
 1500 DO 1700 I=1,NA                                                         ,73
      IA=IA+3                                                                ,74
      CALL BITURP(Q,Q(2),3,NP,KD,A(IA),A(IA+1),NU)                           ,75
      IF (NU)1600,1700,1600                                                  ,76
 1600 NU=NU+10                                                                77
      GO TO 9000                                                             ,78
 1700 CALL TROTPT(T,T(10),1,A(IA),A(IA))                                     ,79    ,80
      GO TO 8000
C
C     NO INTERPOLATION BECAUSE NP .LT. 4 OR CHD .LE. 0 OR PTS. COLLINEAR     ,81
 7000 NA=NP                                                                  ,82
      CALL TRAV(A,P,NP3)
C                                                                            ,83
 8000 NU=MU
C                                                                            ,84
 9000 RETURN                                                                 ,85
      END                                                                    ,86
```

```
          RICHNA                                                       07/31/65
        EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE RICHNA (A,MAX,M,N,P,EPS,CHD,KOD,NA,NU)
      DIMENSION A(3,1),P(1),NU(3),EP(3)
C
C     GIVEN M ARRAYS OF 3D POINTS, WHERE EACH ARRAY HAS N POINTS, STORED
C     IN A SINGLE ARRAY A OF LENGTH MAX.
C     USE RICH3A ON EACH ARRAY SO THAT EACH ARRAY WILL CONTAIN MORE
C     POINTS.  UPON RETURN, THE DENSE ARRAYS WILL BE STORED INA,
C     HEADED BY A TABLE OF CONTENTS.  THE ORIGINAL ARRAYS
C     ARE DESTROYED.
C
      DO 10 I=1,3                                                      ,1
 10   EP(I)=EPS                                                        ,2   ,3
      NA=0                                                             ,4
      MN=M*N                                                           ,5
      IA=MAX-MN                                                        ,6
      IAI=IA+1                                                         ,7
      MAXR=IA-M                                                        ,8
      IDA=M+1                                                          ,9
      IF(KOD)100,100,20                                                ,10
 20   EP(3)=1.E+30
C
C     MOVE ARRAY TO BOTTOM OF A.                                       ,11
 100  DO 200 J=1,MN                                                    ,12
 150  IA=IA+1                                                          ,13
      DO 200 I=1,3                                                     ,14
 200  A(I,IA)=A(I,J)
C                                                                      ,15  ,16  ,17
 300  DO 1000 IB=1,M                                                   ,18
      IF(KOD)305,360,305
C
C     CHECK FOR POINTS WHICH HAVE X,Y NEARLY SAME AS PREV. POINT AND
C     Z DIFFERENT.  WRITE COMMENT IF ANY FOUND.  THESE POINTS WILL BE
C     REMOVED LATER BY NWEED, BUT USER SHOULD KNOW ABOUT IT.
C                                                                      ,19
 305  LTAPE=NU(2)                                                      ,20
      IF (LTAPE)360,360,310                                            ,21
 310  LF=IAI+1                                                         ,22
      LL=IAI+N-1                                                       ,23
      DO 350 L=LF,LL                                                   ,24
      DO 320 I=1,2                                                     ,25
      IF (ABS(A(I,L)-A(I,L-1))-EPS)320,350,350                         ,26
 320  CONTINUE                                                         ,27  ,28
      IF (ABS(A(3,L)-A(3,L-1))-EPS)350,350,330                         ,29
 330  LM=L-1                                                           ,30
      WRITE (LTAPE,340)IB,((A(I,J),I=1,3),J=LM,L)                      ,31  ,32  ,33  ,34  ,35  ,36
                                                                       ,37  ,38  ,39
 340  FORMAT(37HOCHECK THESE 2 POINTS ON PERCENT LINE I4/
     1 8H X Y Z =2(3F10.4,5X))
 350  CONTINUE
C
C     REMOVE POINTS TOO CLOSE TO PRECEEDING POINT                      ,40  ,41
 360  NS=NWEED(A(1,IAI),N,3,EP)                                        ,42
      IF (NS-1)400,400,500
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C     ALL POINTS MUST BE COINCIDENT (WITHIN EP)                              ,43
 400  NU=-1                                                                  ,44
      GO TO 9000
C                                                                            ,45
 500  CALL RICH3A(NS,A(1,IAI),A(1,IAI),KOD,3*MAXR,CHD,ND,A(1,IDA),NU)        ,46
      IF (NU)9000,600,9000
C     MAKE ENTRIES IN TABLE OF CONTENTS                                      ,47
 600  A(1,IB)=P(IB)                                                          ,48
      A(2,IB)=ND                                                             ,49
      A(3,IB)=(IDA-1)*3+1
C         '                                                                  ,50
      IAI=IAI+N                                                              ,51
      IDA=ICA+ND                                                             ,52
      MAXR=MAXR+N-ND                                                         ,53
      NA=NA+ND
C                                                                            ,54
 1000 CONTINUE
C                                                                            ,55    ,56
 9000 RETURN                                                                 ,57
      END                                                                    ,58
```

```
RITE        -  EFN   SOURCE STATEMENT  -  IFN(S)  -                05/22/67

      SUBROUTINE RITE (NFMT,NTAPEO,NP,NROW,NCOL,THETA,U,V,W)
C     PRINT OUT VELOCITY COMPONENTS
      DIMENSION NFMT(1),U(1),V(1),W(1)
      IF (NFMT(1)) 5,999,5
    5 CONTINUE
      IGO=NFMT(1)
      GO TO(10,20,30,40,50,60,70,300,200),IGO
   10 WRITE (NTAPEO,6030)
      GO TO 200
   20 WRITE (NTAPEO,6040)
      GO TO 200
   30 WRITE (NTAPEO,6045)
      GO TO 200
   40 WRITE (NTAPEO,6050)
      GO TO 300
   50 WRITE (NTAPEO,6055)
      GO TO 300
   60 WRITE (NTAPEO,6060)
      GO TO 300
   70 WRITE (NTAPEO,6065)
      GO TO 300

C     PRINT OUT VELOCITY COMPONENTS AT BODY PANELS
  200 CONTINUE
      WRITE (NTAPEO,6010)
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,U)
      WRITE (NTAPEO,6015)
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,V)
      WRITE (NTAPEO,6020)
      CALL OUTB (NTAPEO,NP,NCOL,NROW,THETA,W)
      GO TO 999
C     PRINT OUT VELOCITY COMPONENTS AT WING PANELS
  300 CONTINUE
      WRITE (NTAPEO,6010)
      CALL OUTW (NTAPEO,NP,NCOL,NROW,U)
      WRITE (NTAPEO,6015)
      CALL OUTW (NTAPEO,NP,NCOL,NROW,V)
      WRITE (NTAPEO,6020)
      CALL OUTW (NTAPEO,NP,NCOL,NROW,W)
```

```
6000 FORMAT(1H ,10F11.5)
6010 FORMAT(1H0,5X,8HAXIAL(U))
6015 FORMAT(1H0,5X,13HTRANSVERSE(V))
6020 FORMAT(1H0,5X,11HVERTICAL(W))
6030 FORMAT(1H1, 76H VELOCITY COMPONENTS ON BODY PANELS DUE TO BODY PAN
    1EL PRESSURE SINGULARITIES)
6040 FORMAT(1H1, 76H VELOCITY COMPONENTS ON BODY PANELS DUE TO WING PAN
    1EL PRESSURE SINGULARITIES)
6045 FORMAT(1H1, 55H VELOCITY COMPONENTS ON BODY PANELS DUE TO WING SOU
    1RCES)
6050 FORMAT(1H1, 76H VELOCITY COMPONENTS ON WING PANELS DUE TO WING PAN
    1EL PRESSURE SINGULARITIES)
6055 FORMAT(1H1, 55H VELOCITY COMPONENTS ON WING PANELS DUE TO WING SOU
    1RCES)
6060 FORMAT(1H1, 76H VELOCITY COMPONENTS ON WING PANELS DUE TO BODY PAN
    1EL PRESSURE SINGULARITIES)
6065 FORMAT(1H1, 73H VELOCITY COMPONENTS ON WING PANELS DUE TO BODY LIN
    1E SOURCES AND DOUBLETS)

 999 CONTINUE
     RETURN
     END
```

```
      RRAT       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE RRAT(X,Y,IXY,KODE,XI,YI,C,NU)
      DIMENSION X(1),Y(1),C(3)
C
C     FIND COEFFICIENTS OF QUADRATIC THRU 3 POINTS.
C     1ST PT IS (X,Y), 2ND PT IS (X(IXY+1),Y(IXY+1)),
C     3RD PT IS(X(2*IXY+1),Y(2*IXY+1))
C     IF KODE .GT. 0, EVALUATE AT X=XI
C
      JXY=IXY+32769
      KXY=IXY+JXY
      DX21=X(JXY)-X
      DY21=Y(JXY)-Y
      DX32=X(KXY)-X(JXY)
      DY32=Y(KXY)-Y(JXY)
      DX31=X(KXY)-X
      DY31=Y(KXY)-Y
      SX31=X(KXY)+X
C
 100  IF (DX21*DX32*DX31)300,200,300
 200  NU=1
      GO TO 1000
C
 300  C(3)=(DY32/DX32-DY21/DX21)/DX31
      C(2)=DY31/DX31-C(3)*SX31
      C(1)=Y(JXY)-(C(2)+C(3)*X(JXY))*X(JXY)
      NU=0
C
 400  IF (KODE)1000,1000,500
 500  YI=C(1)+(C(2)+C(3)*XI)*XI
C
 1000 RETURN
      END
```

```
      SUBROUTINE SAVTAP(ISAVET)
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION A(210),B(210),C(210)
      DIMENSION DUM(10),NROW(2)
      IF (ISAVET) 20,20,10
   10 READ (NTAPEC)NBODY,NWING,XMACH,SYM,KACE                                3
      GO TO ( 50, 20, 50),KACE
   20 CALL FSF(6,NTAPEC,IRR)                                                 9
      GO TO 900
   50 NPANEL=NBODY+NWING
      READ (NTAPEC) (I,(DUM(J),J=1,10),I=1,NPANEL),NRG,
     1(NROW(I),I=1,NRG),DUMMY                                               12
      NCOLW=NWING/NROW(NRG)
      CALL FSF( 6,NTAPEC,IRR)                                               32
      GO TO (100,900,200),KACE
  100 CONTINUE
      CALL TTAPE (0,NWING,NWING+NCOLW,NTAPEA,NTAPEC,A,B,C)                 35
      CALL FSF(1,NTAPEA,IRR)                                                37
      CALL TTAPE(0,NWING,NWING,NTAPEA,NTAPEC,A,B,C)                         39
      CALL FSF(1,NTAPEA,IRR)                                                41
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEC,A,B,C)                     43
      CALL FSF(1,NTAPEA,IRR)                                                45
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEC,A,B,C)                     47
      REWIND NTAPEA                                                         48
      CALL TTAPE(1,NWING,NWING+NCOLW,NTAPEB,NTAPEC,A,B,C)                   49
      CALL FSF(1,NTAPEB,IRR)                                                51
      CALL TTAPE(1,NWING,NWING,NTAPEB,NTAPEC,A,B,C)                         53
      REWIND NTAPEB                                                         54
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)                         55
      CALL FSF(1,NTAPEE,IRR)                                                57
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)                         59
      REWIND NTAPEE                                                         60
      GO TO 900
  200 CONTINUE
      CALL TTAPE(0,NPANEL,NWING +NCOLW,NTAPEA,NTAPEC,A,B,C)                 63
      CALL FSF(1,NTAPEA,IRR)                                                65
      CALL TTAPE(0,NWING,NWING,NTAPEA,NTAPEC,A,B,C)                         67
      CALL FSF(1,NTAPEA,IRR)                                                69
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEC,A,B,C)                     71
      CALL FSF(1,NTAPEA,IRR)                                                73
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEC,A,B,C)                     75
      REWIND NTAPEA                                                         76
      CALL TTAPE(1,NPANEL,NWING +NCOLW,NTAPEB,NTAPEC,A,B,C)                 77
      REWIND NTAPEB                                                         78
      CALL TTAPE(1,NBODY,NBODY,NTAPED,NTAPEC,A,B,C)                         79
      CALL FSF(1,NTAPED,IRR)                                                81
      CALL TTAPE(1,NWING,NBODY,NTAPED,NTAPEC,A,B,C)                         83
      CALL FSF(1,NTAPED,IRR)                                                85
      CALL TTAPE(1,NBODY,NWING,NTAPED,NTAPEC,A,B,C)                         87
      CALL FSF(1,NTAPED,IRR)                                                89
      CALL TTAPE(1,NWING,NWING,NTAPED,NTAPEC,A,B,C)                         91
      REWIND NTAPED                                                         92
```

```
      CALL  TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)                    93
      CALL  FSF(1,NTAPEE,IRR)                                           95
      CALL  TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)                    97
      REWIND NTAPEE                                                     98
      CALL  TTAPE(0,NWING,NBODY,NTAPEF,NTAPEC,A,B,C)                    99
      CALL  FSF(1,NTAPEF,IRR)                                           101
      CALL  TTAPE(0,NBODY,NBODY,NTAPEF,NTAPEC,A,B,C)                    103
      CALL  TTAPE(0,NBODY,NWING,NTAPEF,NTAPEC,A,B,C)                    105
      REWIND NTAPEF                                                     106
  900 CONTINUE
      END FILE NTAPEC                                                   107
      REWIND NTAPEC                                                     108
      RETURN
      END
```

SCAMP4 - EFN SOURCE STATEMENT - IFN(S) - 05/22/67

```
      SUBROUTINE SCAMP4(X,Y,N,NDA,NDB,DA,DB,C,S,M)                       SCAM0030
C                                                                        SCAM0020
C                                                                        SCAM0040
C  GIVEN A SET OF  N  POINTS WHOSE ABSCISSAE FORM A STRICTLY MONOTONE    SCAM0050
C  SEQUENCE, AND GIVEN A FIRST OR SECOND DERIVATIVE AT X(1) AND A FIRST  SCAM0060
C  OR SECOND DERIVATIVE AT X(N), TO FIND THE SMOOTHEST POSSIBLE CURVE    SCAM0070
C  PASSING RIGOROUSLY THROUGH THE GIVEN POINTS, SATISFYING THE           SCAM0080
C  SPECIFIED BOUNDARY CONDITIONS, AND POSSESSING CONTINUOUS FIRST AND    SCAM0090
C  SECOND DERIVATIVES.  THE CRITERION OF SMOOTHNESS IS THE               SCAM0100
C  MINIMIZATION OF THE INTEGRAL OF THE SQUARE OF THE SECOND DERIVATIVE,  SCAM0110
C  AND THE CURVE FOUND IS ACCORDINGLY A CHAIN OF CUBICS, I.E., A         SCAM0120
C  SEPARATE CUBIC ON EACH INTERVAL X(I),X(I+1).                          SCAM0130
C                                                                        SCAM0140
      DIMENSION C(4,1),S(1),X(1),Y(1),Z(4)                               SCAM0150
C                                                                        SCAM0170
      L=1                                                                SCAM0180
      KK=1                                                               SCAM0190
      D1=DA                                                              SCAM0200
      D2=DB                                                              SCAM0210
      IF (M-12) 20,10,20                                                 SCAM0220
   10 KK=2                                                               SCAM0230
   20 IF (NDA+1) 30,40,50                                                SCAM0240
   30 D1=DERIV2(X,Y,X)                                                   SCAM0250     9
      GO TO 50                                                           SCAM0260
   40 D1=DERIV1(X,Y,1)                                                   SCAM0270    12
   50 NA=IABS(NDA)                                                       SCAM0280
      IF (NDB+1) 60,70,80                                                SCAM0290
   60 D2=DERIV2(X(N-3),Y(N-3),X(N))                                      SCAM0300    20
      GO TO 80                                                           SCAM0310
   70 D2=DERIV1(X(N-2),Y(N-2),3)                                         SCAM0320    25
   80 NB=IABS(NDB)                                                       SCAM0330
  100 CALL COMCU(D1,D2,S,X,Y,L,M,N,NA,NB)                                SCAM0340    28
      IF (M) 200,110,200                                                 SCAM0350
  110 K=N-1                                                              SCAM0360
      DO 170 J=1,K                                                       SCAM0370
      CALL CUBIC2(X(J),Y(J),S(J),Z,M)                                    SCAM0380    39
      IF (M-1) 120,130,120                                               SCAM0390
  120 M=100*J+M                                                          SCAM0400
      GO TO 200                                                          SCAM0410
  130 GO TO (140,160),KK                                                 SCAM0420
  140 DO 150 I=1,4                                                       SCAM0430
  150 C(I,J)=Z(I)                                                        SCAM0440
      GO TO 170                                                          SCAM0450
  160 L=7*J                                                              SCAM0460
      C(L-6,1)=X(J)                                                      SCAM0470
      C(L-5,1)=X(J+1)                                                    SCAM0480
      C(L-4,1)=3.0                                                       SCAM0490
      C(L-3,1)=Z(1)                                                      SCAM0500
      C(L-2,1)=Z(2)                                                      SCAM0510
      C(L-1,1)=Z(3)                                                      SCAM0520
      C(L  ,1)=Z(4)                                                      SCAM0530
  170 CONTINUE                                                           SCAM0540
      M=0                                                                SCAM0550
  200 RETURN                                                             SCAM0560
      END                                                                SCAM0570
```

```
      SUBROUTINE SIMUN3(X,Y,M,A,K)
      DIMENSION X(1),Y(1)
      DIMENSION C(4,55),FD(55),XI(55),YI(55)

C     NUMERICAL INTEGRATION ROUTINE USING PARABOLIC (OR SIMPSONS)
C     RULE ON DATA FIT WITH CUBIC CHAIN

      K=0
      MS=M
      IF (MS .LE. 2) GO TO 1000

C     USE SCAMP4 TO FIT CUBIC CHAIN TO POINTS
      CALL SCAMP4(X,Y,MS,-1,-1,0.,0.,C,FD,L)
      IF (L .NE. 0) GO TO 1010

      N=MS-1
      IF (MOD(N,2) .EQ. 1) N=N+1
      D=(X(MS)-X(1))/FLOAT(N)

C     DEFINE SET OF M (IF M ODD) OR M+1 (IF M EVEN)
C     EQUALLY SPACED POINTS
      XI(1)=X(1)
      YI(1)=Y(1)
      XI(N+1)=X(MS)
      YI(N+1)=Y(MS)
      JF=1
      DO 200 I=2,N
      XI(I)=XI(I-1)+D
      DO 100 J=JF,MS
      IF (XI(I) .GT. X(J)) GO TO 100
      JF=J-1
      YI(I)=C(1,JF)+XI(I)*(C(2,JF)+XI(I)*(C(3,JF)+C(4,JF)*
     1XI(I)))
      GO TO 200
  100 CONTINUE
      GO TO 1020
  200 CONTINUE

C     USE PARABOLIC RULE TO DO NUMERICAL INTEGRATION
      A=YI(1)+4.*YI(2)+YI(N+1)
      IF (N .LE. 2) GO TO 500
      NL=(N-2)/2
      DO 300 I=1,NL
      A=A+2.*YI(2*I+1)+4.*YI(2*I+2)
  300 CONTINUE

  500 A=D/3.*A
      RETURN

C     SET ERROR CODE
 1000 K=1
      GO TO 500
 1010 K=2

      GO TO 500
 1020 K=3
      GO TO 500
      END
```

```
      SINVRT                                                            08/17/65
   EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C     ROUTINE TO ADJUST STORAGE FOR INV4S AND CALL IT
C
      SUBROUTINE SINVRT(A,K,N,IR1,IR2,SCALE,DET,NDETXP)
C
C     A       = MATRIX TO BE INVERTED
C     K       = ROW DIMENSION STATEMENT ENTRY FOR A
C     N       = NUMBER OF ROWS,COLUMNS IN A
C     IR1     = 0 IF INVERSION SUCCESSFUL
C             = 1 IF OVERFLOW OCCURS
C             = 2 IF MATRIX IS SINGULAR
C             = 3 IF SCALED INVERSE CANNOT BE RE-SCALED
C             = 4 IF ROWS AND COLUMNS CANNOT BE REARRANGED
C                 (MACHINE ERROR)
C             = 10+I IF CODE 3 OCCURRED AFTER CODE I
C             = 20+I IF CODE 4 OCCURRED AFTER CODE I
C     IR2     = CURRENT REDUCTION STAGE, IF IR1=1
C             = RANK, IF IR1=2
C             = 0, OTHERWISE
C     SCALE   = 0, IF IR1 NOT = 3
C             = SCALE FACTOR IF IR1=3
C     DET     = 0, IF IR1=2
C             = DECIMAL PART OF DETERMINANT IF IR1 NOT = 2
C               (A NUMBER GT OR E TO 1.0 AND LT 10.0
C     NDETXP  = 0, IF IR1=2
C             = EXPONENT PART OF DETERMINANT IF IR1 NOT = 2
C               (DETERMINANT= DET*10.0**NDTXP)
C
C     NEEDS N**2+2N+3 CORE LOCATIONS
C
      DIMENSION A(K,10)
C
C     TRANSPOSE TO ROW-WISE STORAGE
C
      NM1=N-1                                                           ,1
      DO 10 J=1,NM1                                                     ,2
      L=J+1                                                             ,3
      DO 10 I=L,N                                                       ,4
      TEMP=A(I,J)                                                       ,5
      A(I,J)=A(J,I)                                                     ,6
 10   A(J,I)=TEMP
C
C     SQEEZE OUT ALL UNUSED STORAGE
      IF (N-K) 25,100,100                                               ,7   ,8   ,9
 25   L=N+1                                                             ,10
      DO 50 J=2,N                                                       ,11
      DO 50 I=1,N                                                       ,12
      A(L,1)=A(I,J)                                                     ,13
 50   L=L+1                                                             ,14
C
C     CALL INVERSION ROUTINE
C
 100  CALL INV4S(A,N,IR1,IR2,SCALE,DET,NDETXP)                          ,15  ,16  ,17
```

```
                SINVRT                                                    08/17/65
           EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

C      IF ANY OTHER ERROR BUT 3, DO NOT REARRANGE
C                                                                    ,18
       IF (IR1) 150,150,125                                          ,19
 125   IF (IR1-3) 7500,150,7500
C
C      EXPAND STORAGE TO NORMAL
C                                                                    ,20
 150   IF (N-K) 175,275,275                                          ,21
 175   L=N**2                                                        ,22
       J=N                                                           ,23
 180   I=N                                                           ,24
 190   A(I,J)=A(L,1)                                                 ,25
       A(L,1)=0.                                                     ,26
       L=L-1                                                         ,27
       IF (I-1) 225,225,200                                          ,28
 200   I=I-1                                                         ,29
       GO TO 190                                                     ,30
 225   IF (J-2)275,275,250                                           ,31
 250   J=J-1                                                         ,32
       GO TO 180
C
C      TRANSPOSE BACK TO COLUMN-WISE STORAGE
C                                                                    ,33
 275   DO 300 J=1,NM1                                                ,34
       L=J+1                                                         ,35
       DO 300 I=L,N                                                  ,36
       TEMP=A(I,J)                                                   ,37
       A(I,J)=A(J,I)                                                 ,38
 300   A(J,I)=TEMP                                                   ,39   ,40   ,41
7500   RETURN                                                        ,42
       END                                                           ,43
```

```
      SUBROUTINE SIZE(MM,NMAX,NPART,NSIZE)
      DIMENSION NSIZE(1)

C     MM     SIZE OF MATRIX
C     NMAX   MAXIMUM SIZE OF PARTITIONS
C     NPART  NUMBER OF PARTITIONS
C     NSIZE  ARRAY OF THE SIZE OF EACH PARTITION

      NPMIN=2
      NT=MM
      NPART=MAXO (NPMIN,1+(NT-1)/NMAX)
      N=NPART
 10   NSIZE(N)=NT/N
      NT=NT-NSIZE(N)
      N=N-1
      IF (N) 20,20,10
 20   RETURN
      END
```

```
      SUBROUTINE SLOPEW
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(16),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION C(4),P(3,50),R(3,50),Q(3,5),LOC(3)
      DIMENSION CLENG(15),NC(3)

C     SUBROUTINE TO CALCULATE WING PANEL THICKNESS
C     AND CAMBER SLOPES
      IF (KOPTF) 500,470,10

C     DEFINE CONSTANTS
C     M,MAX,E AND C ARE CONSTANTS AND ARRAYS
C     USED BY SUBROUTINE PCLXN      IXY,KXY AND NC ARE
C     USED BY SUBROUTINE BITURP                                    ,1
 10   M=3                                                          ,2
      MAX=1                                                        ,3
      E=0.001                                                      ,4
      NPER2=NPER-2                                                 ,5
      NPERT=NPER+1                                                 ,6
      CONST=10000.                                                 ,7
      IXY=1                                                        ,8
      KXY=2                                                        ,9
      ZHT=ZCOR(1,1)                                                ,10
      C(1)=0.                                                      ,11
      C(2)=-1.                                                     ,12
      C(3)=0.                                                      ,13
      NC(1)=0                                                      ,14
      NC(2)=6                                                      ,15
      NC(3)=500

C     DEFINE ARRAYS FOR WING LEADING AND TRAILING
C     EDGES      P IS ARRAY FOR LEADING EDGE AND R
C     IS ARRAY FOR TRAILING EDGE                                   ,16
 20   N1=NPTS(1)                                                   ,17
      DO 30 I=1,N1                                                 ,18    ,19
      P(1,I)=X(I,1)                                                ,20
      P(2,I)=Y(I,1)                                                ,21
 30   P(3,I)=Z(I,1)                                                ,22    ,23
      N2=NPTS(NPER)                                                ,24
      DO 40 I=1,N2                                                 ,25    ,26
      R(1,I)=X(I,NPER)                                             ,27
      R(2,I)=Y(I,NPER)                                             ,28
 40   R(3,I)=Z(I,NPER)                                             ,29    ,30
      IF (KSTART-1) 500,150,50
```

```
C        CALCULATE PANEL SLOPES ON INBOARD CCLUMN IF
C        WING AND BODY INTERSECT IN NON-STREAMWISE LINE                 ,31
   50    I=1                                                            ,32
   60    NUM1=XNUM(1,1)

C        DEFINE ARRAYS FOR AIRFOIL CCORDINATES      XTAB1,ZTAB1
C        ARE ARRAYS FOR UPPER SURFACE COORDINATES     XTAB2,
C        ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES                 ,33
         DO 65 J=1,NUM1                                                 ,34     ,35
         XTAB1(J)=XFOIL(I,J,1)                                          ,36
   65    ZTAB1(J)=ZFOIL(I,J,1)                                          ,37     ,38
         CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                             ,39
         IF (L) 450,70,450                                              ,40
   70    NUM2=XNUM(I,2)                                                 ,41
         DO 75 J=1,NUM2                                                 ,42     ,43
         XTAB2(J)=XFCIL(I,J,2)                                          ,44
   75    ZTAB2(J)=ZFOIL(I,J,2)                                          ,45     ,46
         CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                             ,47
         IF (L) 450,77,450                                              ,48
      77 DO 100 J=1,NPER1

C        CALCULATE CHURD LENGTH
C        XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C        EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C        CLENG IS CHORD LENGTH                                          ,49     ,50
         C(4)=YCON(I,J)                                                 ,51
         CALL POLXN (C,P,M,N1,E,MAX,Q,LOC,NINT)                         ,52
         IF (NINT-1) 480,80,490                                         ,53
   80    XLEAD=Q(1,1)                                                   ,54
         CALL POLXN (C,R,M,N2,E,MAX,Q,LOC,NINT)                         ,55
         IF (NINT-1) 480,90,490                                         ,56
   90    XTRAIL=Q(1,1)                                                  ,57
         CLENG(J)=XTRAIL-XLEAD

C        CALCULATE CAMBER AND THICKNESS ORDINATES
C        ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C        ORDINATES                                                      ,58
         FRAC=(XCON(I,J)-XLEAD)/CLENG(J)                                ,59
         FRAT=(XCEN(I,J)-XLEAD)/CLENG(J)                                ,60
   91    CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)              ,61
         IF (NC(1)) 460,92,92                                           ,62
   92    ZCU=ZR                                                         ,63
         CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)              ,64
         IF (NC(1)) 460,93,93                                           ,65
   93    ZCL=ZR                                                         ,66
         ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(J)+ZHT                          ,67
         CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)              ,68
         IF (NC(1)) 460,97,97                                           ,69
   97    ZTU=ZR                                                         ,70
         CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)              ,71
         IF (NC(1)) 460,98,98                                           ,72
   98    ZTL=ZR                                                         ,73
         ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(J)+ZHT                          ,74
```

```
          SLOPEW                                                      08/12/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

 100  CONTINUE

C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                     ,75    ,76
      IF (NUMS-1) 500,150,103                                      ,77
 103  XTAB1(1)=XLEAD                                               ,78
      ZTAB1(1)=ZFOIL(I,1,1)*CLENG(NPER1)+ZHT                       ,79
      XTAB2(1)=XLEAD                                               ,80
      ZTAB2(1)=ZHT                                                 ,81
      DO 110 J=2,NPER                                              ,82    ,83
      XTAB1(J)=XCON(I,J-1)                                         ,84
      ZTAB1(J)=ZCON(I,J-1)                                         ,85
      XTAB2(J)=XCEN(I,J-1)                                         ,86
 110  ZTAB2(J)=ZCEN(I,J-1)                                         ,87    ,88
      XTAB1(NPERT)=XTRAIL                                          ,89
      ZTAB1(NPERT)=ZFOIL(I,NUM1,1)*CLENG(NPER1)+ZHT                ,90
      XTAB2(NPERT)=XTRAIL                                          ,91
      ZTAB2(NPERT)=ZHT                                             ,92
 120  DO 130 J=2,NPER                                              ,93    ,94
      EPS=CLENG(J-1)/CONST                                         ,95
      XR=XTAB1(J)-EPS                                              ,96
      XS=XTAB1(J)+EPS                                              ,97
 123  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)             ,98
      IF (NC(1)) 460,124,124                                       ,99
 124  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)             ,100
      IF (NC(1)) 460,125,125                                       ,101
 125  ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                           ,102
      XR=XTAB2(J)-EPS                                              ,103
      XS=XTAB2(J)+EPS                                              ,104
 127  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)             ,105
      IF (NC(1)) 460,128,128                                       ,106
 128  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)             ,107
      IF (NC(1)) 460,129,129                                       ,108
 129  THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                           ,109
 130  CONTINUE

C     CALCULATE PANEL SLOPES FOR MAIN WING REGION                  ,110   ,111
 150  DO 250 I=KSTART,KEND

C     CALCULATE CHORD LENGTH
C     XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C     EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C     CLENG IS CHORD LENGTH                                        ,112   ,113
      C(4)=YCCN(I,1)                                               ,114
      CALL POLXN (C,P,M,N1,E,MAX,Q,LOC,NINT)                       ,115
      IF (NINT-1) 480,153,490                                      ,116
 153  XLEAD=Q(1,1)                                                 ,117
      CALL POLXN (C,R,M,N2,E,MAX,Q,LOC,NINT)                       ,118
      IF (NINT-1) 480,157,490                                      ,119
 157  XTRAIL=Q(1,1)                                                ,120
```

```
      CLENG(1)=XTRAIL-XLEAD                                          ,121
      EPS=CLENG(1)/CONST                                             ,122
      IF (NUMS-1) 500,160,170                                        ,123
  160 IR=1                                                           ,124
      GO TO 190                                                      ,125
  170 IR=I+KSTART-1                                                  ,126
  190 NUM1=XNUM(IR,1)
C     DEFINE ARRAYS FOR AIRFOIL COORDINATES     XTAB1,ZTAB1
C     ARE ARRAYS FOR UPPER SURFACE COORDINATES     XTAB2,
C     ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES                 ,127
      DO 195 J=1,NUM1                                                ,128   ,129
      XTAB1(J)=XFOIL(IR,J,1)                                         ,130
  195 ZTAB1(J)=ZFOIL(IR,J,1)                                         ,131   ,132
      CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                             ,133
      IF (L) 450,200,450                                             ,134
  200 NUM2=XNUM(IR,2)                                                ,135
      DO 205 J=1,NUM2                                                ,136   ,137
      XTAB2(J)=XFOIL(IR,J,2)                                         ,138
  205 ZTAB2(J)=ZFOIL(IR,J,2)                                         ,139   ,140
      CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                             ,141
      IF (L) 450,207,450
C     CALCULATE CAMBER AND THICKNESS ORDINATES
C     ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C     ORDINATES                                                      ,142
  207 DO 220 J=1,NPER1                                               ,143   ,144
      FRAC=(XCON(I,J)-XLEAD)/CLENG(1)                                ,145
      FRAT=(XCEN(I,J)-XLEAD)/CLENG(1)                                ,146
  210 CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)              ,147
      IF (NC(1)) 460,211,211                                         ,148
  211 ZCU=ZR                                                         ,149
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)              ,150
      IF (NC(1)) 460,212,212                                         ,151
  212 ZCL=ZR                                                         ,152
      ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(1)+ZHT                          ,153
      CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)              ,154
      IF (NC(1)) 460,217,217                                         ,155
  217 ZTU=ZR                                                         ,156
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)              ,157
      IF (NC(1)) 460,218,218                                         ,158
  218 ZTL=ZR                                                         ,159
      ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(1)+ZHT                          ,160
  220 CONTINUE
C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                       ,161   ,162
      IF (I-KSTART) 500,230,223                                      ,163
  223 IF (NUMS-1) 500,225,230                                        ,164
  225 K=KSTART                                                       ,165
```

```
      DO 227 J=1,NPER1                                          ,166  ,167
      ALPHA(I,J)=ALPHA(K,J)                                     ,168
  227 THETA(I,J)=THETA(K,J)                                     ,169  ,170
      GO TO 250                                                 ,171
  230 XTAB1(1)=XLEAD                                            ,172
      ZTAB1(1)=ZFOIL(IR,1,1)*CLENG(1)+ZHT                       ,173
      XTAB2(1)=XLEAD                                            ,174
      ZTAB2(1)=ZHT                                              ,175
      DO 231 J=2,NPER                                           ,176  ,177
      XTAB1(J)=XCON(I,J-1)                                      ,178
      ZTAB1(J)=ZCON(I,J-1)                                      ,179
      XTAB2(J)=XCEN(I,J-1)                                      ,180
  231 ZTAB2(J)=ZCEN(I,J-1)                                      ,181  ,182
      XTAB1(NPERT)=XTRAIL                                       ,183
      ZTAB1(NPERT)=ZFOIL(IR,NUM1,1)*CLENG(1)+ZHT                ,184
      XTAB2(NPERT)=XTRAIL                                       ,185
      ZTAB2(NPERT)=ZHT                                          ,186
      DO 240 J=2,NPER                                           ,187  ,188
      XR=XTAB1(J)-EPS                                           ,189
      XS=XTAB1(J)+EPS                                           ,190
  233 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)          ,191
      IF (NC(1)) 460,234,234                                    ,192
  234 CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)          ,193
      IF (NC(1)) 460,235,235                                    ,194
 235  ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                        ,195
      XR=XTAB2(J)-EPS                                           ,196
      XS=XTAB2(J)+EPS                                           ,197
  237 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)          ,198
      IF (NC(1)) 460,238,238                                    ,199
  238 CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)          ,200
      IF (NC(1)) 460,239,239                                    ,201
 239  THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                        ,202
  240 CONTINUE                                                  ,203  ,204
  250 CONTINUE                                                  ,205  ,206
      IF (KSTART-1) 500,257,253                                 ,207
 253  IF (NUMS-1) 500,254,257                                   ,208
  254 K=KSTART                                                  ,209
      DO 255 J=1,NPER1                                          ,210  ,211
      ALPHA(1,J)=ALPHA(K,J)                                     ,212
  255 THETA(1,J)=THETA(K,J)
C
C     CALCULATE PANEL SLOPES FOR WING TIP REGION
C     IF OUTBOARD WING EDGE IS NON-STREAMWISE LINE              ,213  ,214
 257  IF (KEND-NPLN1) 258,510,500                               ,215
  258 I=NPLN1                                                   ,216
      IF (SLOPE) 260,500,280                                    ,217
  260 DO 270 J=1,NPER                                           ,218  ,219
      JN=NPER+1-J                                               ,220
      R(1,J)=XCOR(I+1,JN)                                       ,221
      R(2,J)=YCOR(I+1,JN)                                       ,222
  270 R(3,J)=ZCOR(I+1,JN)                                       ,223  ,224
      GO TO 310                                                 ,225
  280 DO 290 J=1,NPER                                           ,226  ,227
      P(1,J)=XCOR(I+1,J)                                        ,228
```

```
      P(2,J)=YCOR(I+1,J)                                       ,229
  290 P(3,J)=ZCOR(I+1,J)                                       ,230   ,231
  310 NUM1=XNUM(NUMS,1)

C     DEFINE ARRAYS FOR AIRFOIL COORDINATES     XTAB1,ZTAB1
C     ARE ARRAYS FOR UPPER SURFACE COORDINATES      XTAB2,
C     ZTAB2 ARE ARRAYS FOR LOWER SURFACE COORDINATES           ,232
      DO 315 J=1,NUM1                                          ,233   ,234
      XTAB1(J)=XFOIL(NUMS,J,1)                                 ,235
  315 ZTAB1(J)=ZFOIL(NUMS,J,1)                                 ,236   ,237
      CALL CHECK (XTAB1,NUM1,2,L,NWRITE)                       ,238
      IF (L) 450,320,450                                       ,239
  320 NUM2=XNUM(NUMS,2)                                        ,240
      DO 325 J=1,NUM2                                          ,241   ,242
      XTAB2(J)=XFOIL(NUMS,J,2)                                 ,243
  325 ZTAB2(J)=ZFOIL(NUMS,J,2)                                 ,244   ,245
      CALL CHECK (XTAB2,NUM2,2,L,NWRITE)                       ,246
      IF (L) 450,327,450                                       ,247
  327 DO 380 J=1,NPER1

C     CALCULATE CHORD LENGTH
C     XLEAD AND XTRAIL ARE WING LEADING- AND TRAILING-
C     EDGE INTERCEPTS OF CHORD THROUGH CONTROL POINT
C     CLENG IS CHORD LENGTH                                    ,248   ,249
      C(4)=YCCN(I,J)                                           ,250
      IF (SLOPE) 330,500,340                                   ,251
  330 NL=N1                                                    ,252
      NT=NPER                                                  ,253
      GO TO 350                                                ,254
  340 NL=NPER                                                  ,255
      NT=N2                                                    ,256
  350 CALL POLXN (C,P,M,NL,E,MAX,Q,LOC,NINT)                   ,257
      IF (NINT-1) 480,360,490                                  ,258
  360 XLEAD=Q(1,1)                                             ,259
      IF (J-1) 500,363,364                                     ,260
  363 XLS=XLEAD                                                ,261
  364 CALL POLXN (C,R,M,NT,E,MAX,Q,LOC,NINT)                   ,262
      IF (NINT-1) 480,367,490                                  ,263
  367 XTRAIL=Q(1,1)                                            ,264
      IF (J-NPER1) 370,368,500                                 ,265
  368 XTS=XTRAIL                                               ,266
      CLS=XTS-XLS                                              ,267
  370 CLENG(J)=XTRAIL-XLEAD

C     CALCULATE CAMBER AND THICKNESS ORDINATES
C     ZCON AND ZCEN ARE Z-CAMBER AND Z-THICKNESS
C     ORDINATES                                                ,268
      FRAC=(XCON(I,J)-XLEAD)/CLENG(J)                          ,269
      FRAT=(XCEN(I,J)-XLEAD)/CLENG(J)                          ,270
  371 CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAC,ZR,NC)        ,271
      IF (NC(1)) 460,372,372                                   ,272
  372 ZCU=ZR                                                   ,273
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAC,ZR,NC)        ,274
      IF (NC(1)) 460,373,373                                   ,275
```

```
 373  ZCL=ZR                                                      ,276
      ZCON(I,J)=((ZCU+ZCL)/2.)*CLENG(J)+ZHT                       ,277
      CALL BITURP (XTAB1,ZTAB1,IXY,NUM1,KXY,FRAT,ZR,NC)           ,278
      IF (NC(1)) 460,377,377                                      ,279
 377  ZTU=ZR                                                      ,280
      CALL BITURP (XTAB2,ZTAB2,IXY,NUM2,KXY,FRAT,ZR,NC)           ,281
      IF (NC(1)) 460,378,378                                      ,282
 378  ZTL=ZR                                                      ,283
      ZCEN(I,J)=((ZTU-ZTL)/2.)*CLENG(J)+ZHT                       ,284
  380 CONTINUE

C     CALCULATE CAMBER AND THICKNESS SLOPES
C     ALPHA AND THETA ARE ALPHA-CAMBER AND -THICKNESS
C     SLOPES
C     NOTE THAT IF SINGLE AIRFOIL IS INPUT ON
C     CARDS CONSTANT ARRAY FOR ALPHA-CAMBER AND
C     -THICKNESS IS CALCULATED                                    ,285   ,286
      IF (NUMS-1) 500,383,387                                     ,287
  383 K=KSTART                                                    ,288
      L=KEND                                                      ,289
      DO 385 J=1,NPER1                                            ,290   ,291
      ALPHA(L,J)=ALPHA(K,J)                                       ,292
  385 THETA(L,J)=THETA(K,J)                                       ,293   ,294
      GO TO 510                                                   ,295
  387 XTAB1(1)=XLS                                                ,296
      ZTAB1(1)=ZFOIL(NUMS,1,1)*CLS+ZHT                            ,297
      XTAB2(1)=XLS                                                ,298
      ZTAB2(1)=ZHT                                                ,299
      DO 390 J=2,NPER                                             ,300   ,301
      XTAB1(J)=XCON(I,J-1)                                        ,302
      ZTAB1(J)=ZCON(I,J-1)                                        ,303
      XTAB2(J)=XCEN(I,J-1)                                        ,304
 390  ZTAB2(J)=ZCEN(I,J-1)                                        ,305   ,306
      XTAB1(NPERT)=XTS                                            ,307
      ZTAB1(NPERT)=ZFOIL(I,NUM1,1)*CLS+ZHT                        ,308
      XTAB2(NPERT)=XTS                                            ,309
      ZTAB2(NPERT)=ZHT                                            ,310
 400  DO 410 J=2,NPER                                             ,311   ,312
      EPS=CLENG(J-1)/CONST                                        ,313
      XR=XTAB1(J)-EPS                                             ,314
      XS=XTAB1(J)+EPS                                             ,315
 403  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XR,ZR,NC)            ,316
      IF (NC(1)) 460,404,404                                      ,317
 404  CALL BITURP (XTAB1,ZTAB1,IXY,NPERT,KXY,XS,ZS,NC)            ,318
      IF (NC(1)) 460,405,405                                      ,319
 405  ALPHA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                          ,320
      XR=XTAB2(J)-EPS                                             ,321
      XS=XTAB2(J)+EPS                                             ,322
 407  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XR,ZR,NC)            ,323
      IF (NC(1)) 460,408,4C8                                      ,324
 408  CALL BITURP (XTAB2,ZTAB2,IXY,NPERT,KXY,XS,ZS,NC)            ,325
      IF (NC(1)) 460,409,409                                      ,326
 409  THETA(I,J-1)=ATAN((ZS-ZR)/(XS-XR))                          ,327
  410 CONTINUE                                                    ,328   ,329
```

```
 420 GO TO 510
C      ERROR MESSAGES                                   ,330
 450 WRITE (NWRITE,940)                                 ,331  ,332
     GO TO 495                                          ,333
 460 WRITE (NWRITE,930)                                 ,334  ,335
     GO TO 495                                          ,336
 470 WRITE (NWRITE,900)                                 ,337  ,338
     GO TO 495                                          ,339
 480 WRITE (NWRITE,910)                                 ,340  ,341
     GO TO 495                                          ,342
 490 WRITE (NWRITE,920)                                 ,343  ,344
     GO TO 495                                          ,345
 495 KODEW=1                                            ,346
     GO TO 500                                          ,347
 500 RETURN

C      USE SUBROUTINE MEAN TO SMOOTH CAMBER AND
C      THICKNESS SLOPES                                 ,348
 510 IF (NUMS-1) 500,520,550                            ,349
 520 K=KSTART                                           ,350
     DO 530 J=1,NPER1                                   ,351  ,352
     XTAB1(J)=XCON(K,J)                                 ,353
     ZTAB1(J)=ALPHA(K,J)                                ,354
     XTAB2(J)=XCEN(K,J)                                 ,355
 530 ZTAB2(J)=THETA(K,J)                                ,356  ,357
 533 CALL MEAN (XTAB1,ZTAB1,NPER1)                      ,358
 537 CALL MEAN (XTAB2,ZTAB2,NPER1)                      ,359
     DO 540 I=1,NPLN1                                   ,360  ,361
     DO 540 J=1,NPER1                                   ,362  ,363
     ALPHA(I,J)=ZTAB1(J)                                ,364
 540 THETA(I,J)=ZTAB2(J)                                ,365  ,366  ,367
     GO TO 500                                          ,368
 550 DO 580 I=1,NPLN1                                   ,369  ,370
     DO 560 J=1,NPER1                                   ,371  ,372
     XTAB1(J)=XCON(I,J)                                 ,373
     ZTAB1(J)=ALPHA(I,J)                                ,374
     XTAB2(J)=XCEN(I,J)                                 ,375
 560 ZTAB2(J)=THETA(I,J)                                ,376  ,377
 563 CALL MEAN (XTAB1,ZTAB1,NPER1)                      ,378
 567 CALL MEAN (XTAB2,ZTAB2,NPER1)                      ,379
     DO 570 J=1,NPER1                                   ,380  ,381
     ALPHA(I,J)=ZTAB1(J)                                ,382
 570 THETA(I,J)=ZTAB2(J)                                ,383  ,384
 580 CONTINUE                                           ,385  ,386
     GO TO 500                                          ,387
 900 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
    119HPROGRAM LOGIC ERROR/10X,
    234HINCORRECT COMPUTED GO TO STATEMENT)
 910 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
    124HSUBROUTINE POLXN FAILURE/10X,
    255HWING CHORD LENGTH NOT CALCULATED  NO INTERSECTION FOUND)
 920 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
    124HSUBROUTINE POLXN FAILURE/10X,
```

```
      SLOPEW                                                   08/12/65
    EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

   262HWING CHORD LENGTH NOT CALCULATED  MULTIPLE INTERSECTIONS FOUND)
930 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
   125HSUBROUTINE BITURP FAILURE/10X,
   233HINTERPOLATED VALUE NOT CALCULATED)
940 FORMAT(1H1,9X,33HERROR MESSAGE - SUBROUTINE SLOPEW/10X,
   121HINPUT CARD DATA ERROR/10X,
   147HX-COORDINATES OF AIRFOIL NOT STRICTLY MONOTONIC/10X,
   219HINCREASING SEQUENCE)
     END                                                          ,388
```

```
      SUBROUTINE STRM1(X,Y,Z,XMAX,XMIN,XDELT,IND1)
      COMMON DUM(4),N8,DM(3),N5,N6
      COMMON /TODO/ DXMAX,DXMIN,PRINT
      COMMON /INTGA/A(875),B(875),C(875),D(875)
      COMMON /JDED/JCD
      DIMENSION NU(3)
      DIMENSION TD(45)
      DIMENSION XT(200),YT(200),ZT(200),CPT(200)
      EQUIVALENCE (A,XT),(B,YT),(C,ZT),(D,CPT)
      EQUIVALENCE (NUM1,TD(1))
 1014 FORMAT(1H0,7X,1HX,14X,1HY,14X,1HZ,13X,3HCPL,/1X,F15.3,3F15.4)
 1015 FORMAT(1X,F15.3,3F15.4)
 1016 FORMAT(1H0,38HACTUAL POINTS DETERMINED BY INTEGRATOR/,7X,1HX,14X,1
     $HY,14X,1HZ,14X,1HU,14X,1HV,14X,1HW,13X,2HCP)
 1017 FORMAT(1X,7F15.6)
      IF(XDELT.LT..001) XDELT=.001
C          IND1 INDICATES WHICH WAY THE STREAMLINE INTEGRATION TAKES PLACE
C
C
C          HAVING THE SINGULARITY STRENGTHS PREVIOUSLY CALCULATED
           FSGN=(-1)**IND1                                          4
           NUM1=3
           NUM=3
      IND=0
      I=0
      CALL LACKEY(1,X,Y,Z,U,V,W,CP)                                 9
      CPLT=CP
           ENM=SQRT(U*U+V*V+W*W)*FSGN                              11
           TD(36)=X
           TD(35)=Y
           TD(34)=Z
           TD(33)=U/ENM
           TD(32)=V/ENM
           TD(31)=W/ENM
      TD(37)=XDELT
      TD(38)=0.
      TD(40)=.5
      TD(41) =DXMIN
      TD(42)=DXMAX
      TD(43)=1.
      TD(44)=1000.
      TD(45)=1.E-02
      JC=0
      CALL DES(NUM1,IND,NTEST)                                     13
   10 IF(NTEST) 20,20,15
   15 CALL DEI(NUM,IND,NTEST)                                      17
   20 CALL LACKEY(1,TD(36),TD(35),TD(34),U,V,W,CP)                 19
      ENM=SQRT(U*U+V*V+W*W)*FSGN                                   21
      TD(33)=U/ENM
      TD(32)=V/ENM
      TD(31)=W/ENM
      CALL DED(NARG)                                               22
      IF(NARG.EQ.0) GO TO 10
```

```
      IF(JCD.EQ.5) GO TO 35
      JC=0
   29 I=I+1
      XT(I)=TD(36)
      YT(I)=TD(35)
      ZT(I)=TD(34)
      CPT(I)=CP
      IF(JCD.EQ.5) GO TO 15
      IF(I.LT.200.AND.XT(I).LT.XMAX.AND.XT(I).GT.XMIN) GO TO 15
      IF(IND1.EQ.2) GO TO 38
      IS=I/2
      DO 36 J=1,IS
      L=I-J+1
      TA=A(L)
      TB=B(L)
      TC=C(L)
      TDD=D(L)
      A(L)=A(J)
      B(L)=B(J)
      C(L)=C(J)
      D(L)=D(J)
      A(J)=TA
      B(J)=TB
      C(J)=TC
      D(J)=TDD
   36 CONTINUE
      IF(PRINT.NE.0.) GO TO 60
      FIN=AMOD(XT(1),XDELT)
      XINI=XT(1)+XDELT-FIN
       IXM=ABS(XT(I)-XINI)/XDELT
      XL=XINI
      NU(1)=0
      NU(2)=6
      NU(3)=0
      CALL BJTURP(XT,YT,1,I,2,XL,YL,NU)                    76
      CALL BJTURP(XT,ZT,1,I,2,XL,ZL,NU)                    78
      CALL BJTURP(XT,CPT,1,I,2,XL,CPL,NU)                  80
      WRITE(N6,1014) XL,YL,ZL,CPL                          81
      GO TO 51
   38 CONTINUE
      XL=X-XDELT
      DO 40 J=1,I
      K=I-J+2
      L=K-1
      XT(K)=XT(L)
      YT(K)=YT(L)
      ZT(K)=ZT(L)
      CPT(K)=CPT(L)
   40 CONTINUE
      I=I+1
      XT(1)=X
      YT(1)=Y
      ZT(1)=Z
      CPT(1)=CPLT
```

```
      IF(PRINT.NE.0.) GO TO 65
       IXM=ABS(XT(I)-XT(1))/XDELT
   51 CONTINUE
      DO 52 J=1,IXM
      XL=XL+XDELT
      CALL BJTURP(XT,YT,1,I,2,XL,YL,NU)                                  109
      CALL BJTURP(XT,CPT,1,I,2,XL,CPL,NU)                                111
      CALL BJTURP(XT,ZT,1,I,2,XL,ZL,NU)                                  113
      WRITE(N6,1015) XL,YL,ZL,CPL                                        114
   52 CONTINUE
   53 CONTINUE
      RETURN
   35 JC=JC+1
      IF(JC.LE.3)  GO TO 29
      I=I-JC+1
      JC=1
      GO TO 29
   60 CONTINUE
      WRITE(N6,1016)                                                     126
      DO 64 J=1,I
      CALL LACKEY(1,XT(J),YT(J),ZT(J),U,V,W,CP)                          133
   64 WRITE(N6,1017) XT(J),YT(J),ZT(J),U,V,W,CP                          134
      GO TO 53
   65 CONTINUE
      DO 67 J=1,I
      CALL LACKEY(1,XT(J),YT(J),ZT(J),U,V,W,CP)                          148
   67 WRITE(N6,1017) XT(J),YT(J),ZT(J),U,V,W,CP                          149
      GO TO 53
      END
```

```
        TDUMP                                                          09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TDUMP(LO,LTAPE,LGDEF)
      DIMENSION R(2000),IR(10)
      LOGICAL LGDEF(3,6)
C
C     USED BY GEOMD TO DUMP CONTENTS OF PERCENT LINE TAPE
C
      DO 10 M=1,6                                              ,1
      IF(LGDEF(3,M)) GO TO 50                                  ,2   ,3   ,4
 10   CONTINUE                                                 ,5   ,6
      GO TO 9000                                               ,7
C
 50   WRITE(LO,100)                                            ,8   ,9
 100  FORMAT(10H1TAPE CUMP)
      REWIND LTAPE                                             ,10
      JR=0                                                     ,11
      DO 1000 K=1,9                                            ,12
      JR=JR+1                                                  ,13
      READ (LTAPE) IR                                          ,14  ,15  ,16
      WRITE(LO,200)JR,IR                                       ,17  ,18  ,19
 200  FORMAT(7HORECORD I3/10I6)
      JR=JR+1                                                  ,20
      L=IR(2)                                                  ,21
      READ(LTAPE)(R(I),I=1,L)                                  ,22  ,23  ,24  ,25  ,26
      WRITE(LO  ,300)JR,(R(I),I=1,L)                           ,27  ,28  ,29  ,30  ,31  ,32

 300  FORMAT(7HORECORD I3/(1H 9F13.4))
 1000 CONTINUE                                                 ,33  ,34
 9000 RETURN                                                   ,35
      END                                                      ,36
```

```
      SUBROUTINE TFLAT(DAT,LI,LO,LA,LB,LD,NBODY,IOK)
C
C     USED BY OPCAM TO TRANSFORM BODY AND WING TO A NEW
C     COORDINATE SYSTEM, FLATTEN THE WING, AND INTERSECT
C     THE TRANSFORMED BODY WITH THE TRANSFORMED AND FLATTENED WING
C
      DIMENSION DAT(2),STA(51),OLDSTA(51),
     , B(4368),RT(3,4),RX(3,51),W(4368),NU(3),BCODE(2),S(1000),
     , BW(3,100),ALFBET(2),NAME(3),RZ(51,2)
     ,,IREC(10),JREC(1C),DUM(9)
     ,,BTITL(12),WTITL(12),BAX(2)
      LOGICAL LGDEF,BODY,WING
      COMMON /LGEOMD/LGDEF(3,6)
      DATA IREC,JREC,DUM,NU/10*0,1,9,1,7*0,9*0.,0,6, 30/,
     , NAME/6HDEFINE,6HDEFINE,6HEND OF/
     , ,BAX/0.,0./,BTITL(1)/
     ,72HTRANSFORMED BODY (PRIMED-AXIS SYSTEM)
     ,           /,WTITL(1)/
     ,72HPOINTS ON FLAT TRANSFORMED WING (PRIMED-AXIS SYSTEM)
     ,           /
C INPUTS ARE
C     DAT = DATE
C     LI = INPUT TAPE
C     LO = OUTPUT TAPE
C     LA = TAPE WRITTEN BY GEOMETRIC DEFINITION LINK, AND
C          USED BY TFLAT
C     LB = TAPE TO BE WRITTEN FOR AERO LINK
C     LD = TAPE TO BE WRITTEN FOR PANEL LINK
C OUTPUT IS
C     IOK = 0 IF TFLAT IS SUCCESSFUL
C
 100  WRITE(LO,110)DAT                                               ,1    ,2    ,3
 110  FORMAT(1H1 55X 2A6/68H PROGRAM ENTERED LINK TO TRANSFORM BODY AND/
     ,OR WING BEFORE PANELLING)

C     CLEAR STORAGE (PROGRAM DOES NOT REQUIRE THIS, BUT HELPFUL FOR DUMPS)
      DO 120 I=1,4368                                                ,4
      B(I)=-0.                                                       ,5
 120  W(I)=-0.                                                       ,6    ,7
      CALL TRAV(RX,B,153)                                            ,8
      CALL TRAV(S,B,10C0)                                            ,9
      CALL TRAV(BW,B,3C0)                                            ,10
      CALL TRAV(RZ,B,1C2)
                                                                     ,11
      NU(2)=LO                                                       ,12
      NBODY=0                                                        ,13
      REWIND LA                                                      ,14
      REWIND LD                                                      ,15
      READ(LI,210) XNRX,CHD,BCODE,EPRC                               ,16   ,17   ,18
 210  FORMAT(6F10.0)
      NRX=XNRX                                                       ,19
      KODEBR=BCODE(1)                                                ,20
      KODEBS=BCODE(2)
```

```
          TFLAT                                                      09/14/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C     NRX = NO. OF POINTS DESIRED IN RX ARRAY (RADIUS AND Z VS. X)
C       LINEAR INTERPOLATION, =2 FOR BIQUADRATIC INTERPOLATION.
C     KODEBR IS USED BY BITURP TO GET RX ARRAY
C     KODEBS IS USED BY WBXX
C     CHD IS USED BY TFLAT1
C     EPRC IS USED BY TFLAT1.  Z-CENTROID AND RADIUS SET =0 IF LESS THAN EPRC.
C                                                                         ,21
      MU = 2                                                              ,22
      IF(NRX .GT. 2 .AND. NRX .LT. 50) GO TO 400                          ,23    ,24    ,25
      NRX=50                                                              ,26
 400  CONTINUE
C
C     TEST FOR ERROR IN PREVIOUS LINK                                     ,27
      BODY=.FALSE.                                                        ,28
      WING=.FALSE.                                                        ,29
      IF(.NOT. LGDEF(1,3)) GO TO 500
C     BODY WAS REQUESTED.  SEE IF PROCESSED.                              ,30    ,31    ,32
      IF(.NOT. LGDEF(3,3)) GO TO 700                                      ,33    ,34    ,35
      BODY= .TRUE.
C     CHECK WING                                                          ,36
 500  IF(.NOT. LGDEF(1,1)) GO TO 600
C     UPPER WING REQUESTED.  SEE IF PROCESSED.                            ,37    ,38    ,39
      IF(.NOT. LGDEF(3,1)) GO TO 700                                      ,40    ,41    ,42
      WING= .TRUE.
C     CHECK THAT AT LEAST A BODY OR A WING IS DEFINED                     ,43
 600  IF(BODY .OR. WING) GO TO 800
C                                                                         ,44    ,45    ,46
 700  WRITE(LO,710)                                                       ,47    ,48
 710  FORMAT(23HOERROR IN PREVIOUS LINK)
      GO TO 8000
C                                                                         ,49
 800  IF(BODY) GO TO 1C00
C     NO BODY, BUT THERE IS A WING.  WRITE DUMMY RECORDS FOR BODY
C       MERIDIAN LINES ON TAPE LD.                                        ,50    ,51    ,52
      WRITE(LD)JREC                                          LD 1         ,53    ,54    ,55
      WRITE(LD)DUM                                           LD 2
C     SKIP RECORDS 1-4 ON TAPE LA                                         ,56    ,57    ,58
      CALL FSR(4,LA,IOK)                                     LA 1-4       ,59
      GO TO 3000
C
C     READ BODY DEFINITION FROM TAPE LA                                   ,60
 1000 CONTINUE                                                            ,61
      READ(LA)IREC                                           LA 1         ,62    ,63    ,64
      NSTA=IREC(2)                                                        ,65
      MU = 4                                                              ,66
      IF((NSTA .LT. 2) .OR. (NSTA .GT. 51) .OR. (IREC(1) .NE. 1) .OR.
     , (IREC(3) .NE. 0)) GO TO 8000                                       ,67    ,68    ,69
      READ(LA)(STA(I),I=1,NSTA)                              LA 2
C     SAVE ORIGINAL BODY STATIONS FOR LATER PRINTOUT                      ,70    ,71    ,72    ,73    ,74
      CALL TRAV(OLDSTA,STA,NSTA)                                          ,75
      READ(LA)IREC                                           LA 3         ,76    ,77    ,78
      MU = 6                                                              ,79
      IF(IREC(1) .NE. ? .OR. IREC(3) .NE. 0 .OR. IREC(2) .GT. 4368)
     , GO TO 8000                                                         ,80    ,81    ,82
```

```
          EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      NBPCT=IREC(4)
C     NBPCT = NO. OF BODY MERIDIAN LINES                                   ,83
      NBEL=IREC(2)                                                         ,84
      READ(LA)(B(I),I=1,NBEL)                               LA 4
C
C     FIND TRANSFORMATION RT TO NEW COORD. SYSTEM.
C     TRANSFORM AND SHEAR BODY MERIDIAN LINES B
C     TRANSFORM BODY STATIONS
C     FIND RADIUS AND Z-COORD. OF CENTROID VS. X-PRIME                     ,85   ,86   ,87   ,88   ,89
 2000 MU = 8                                                               ,90
      CALL TFLAT1(CHD,NSTA,STA,NBPCT,B,S,W,RT,ALFBET,RZ,EPRC,NU)           ,91
      IF(NU)8000,2050,8000                                                 ,92
 2050 CONTINUE
C                                                                          ,93
      WRITE(LO,2100)(RT(I,4),I=1,3),(RT(I,1),I=1,3),
     , (I,OLDSTA(I),STA(I),RZ(I,1),RZ(I,2),I=1,NSTA)                      ,94   ,95   ,96   ,97   ,98   ,99
                                                                           ,100  ,101  ,102  ,103  ,104
 2100 FORMAT(51HOTHE ORIGIN OF THE NEW BODY COORDINATE SYSTEM IS AT /
     , 5X12H(X,Y,Z) =  ( F10.4,1H,F10.4,1H,F10.4,1H)/
     , 52HOA UNIT VECTOR ALONG THE X-PRIME AXIS HAS COMPONENTS /
     , 5X12H(X,Y,Z) =  ( F10.4,1H,F10.4,1H,F10.4,1H)/
     , /1HO 12X 8HORIGINAL 5X 12HCIRCULARIZED 7X 4HBODY 8X 6HZ-BODY
     , /6X 3HNO. 2X 12HBODY STATION 3X 12HBODY STATION 6X 6HRADIUS
     , 6X 8HCENTROID//(1H I7,F13.4,F15.4,2F14.4))
C
C     WRITE TRANSFORMED BODY MERIDIAN LINES ON OUTPUT TAPE
      CALL TFLATM(B,NBPCT,BAX,BTITL,DAT,LO,STA)
C
C     WRITE BODY RECORDS ON TAPE LD                                        ,105
      IREC(1)=1                                                            ,106
      WRITE(LD)IREC                                         LD 1           ,107  ,108  ,109
      WRITE(LD)(B(I),I=1,NBEL)                              LD 2
C
C     BODY DONE.  PROCESS WING.
C                                                                          ,110  ,111  ,112  ,113  ,114
      IF(WING) GO TO 3C00
C     NO. WING.  WRITE DUMMY WING RECORDS ON TAPE LD                       ,115  ,116  ,117
      JREC(3)=1                                                            ,118
      DO 2500 I=3,5,2                                                      ,119
      JREC(1)=I                                                            ,120
      WRITE(LD)JREC                                         LD 3,5         ,121  ,122  ,123
 2500 WRITE(LD)DUM                                          LD 4,6         ,124  ,125  ,126  ,127
      ZAV=0.                                                               ,128
      GO TO 5000
C
C     WING IS TO BE PRCCESSED.  READ WING DEFINITION FROM TAPE LA          ,129
 3000 CONTINUE                                                             ,130
      CALL FSR(2,LA,IOK)                                    LA 5-6         ,131
      READ(LA)IREC                                          LA 7           ,132  ,133  ,134
      MU = 10                                                              ,135
      IF(IREC(1) .NE. 7 ) GO TO  8000                                      ,136  ,137  ,138
      NWPCL=IREC(4)                                                        ,139
      NWEL=IREC(2)
C     NWPCL = NO. OF WING PERCENT LINES                                    ,140
```

```
      MU = 12                                                          ,141
      IF(NWEL .GT. 4368) GO TO 8000
C     RECORD 3 ON LD IS SAME AS 7 ON LA (EX. WORD 1).  WRITE IT.      ,142  ,143  ,144
      IREC(1)=3                                                        ,145
      WRITE(LD)IREC                                          LD 3
C                                                                      ,146  ,147  ,148
      READ(LA) (W(I),I=1,NWEL)                               LA 8
C
C     WE DO NOT TRANSFORM WING IF THERE IS NO BODY                    ,149  ,150  ,151  ,152  ,153
      IF(.NOT. BODY) GO TO 4000
C
C     TRANSFORM WING                                                   ,154  ,155  ,156
 3050 CALL TRAPCT(W,NWPCL,RT)
C
C     FIND WING-BODY INTERSECTIONS                                     ,157
 3100 MU = 14                                                          ,158
      CALL WBXX(B,NBPCT,W,NWPCL,1.E-4,KODEBS,S,BW,NU)                  ,159
      IF(NU)8000,3200,8000
C     INTERSECTION POINTS ARE IN BW.  USE AVERAGE OF Z FOR
C     FIRST AND LAST INTERSECTION POINTS.                              ,160
 3200 ZAV=(BW(3,1)+BW(3,NWPCL))*.5
C     CHECK THAT BOTH INTERSECTION POINTS WERE OK.  IF ANY WING
C     PERCENT LINE FAILED TO PIERCE BODY, THEN COORD. OF 1.E+30
C     WAS INSERTED (BY WBX) FOR X,Y AND Z.                             ,161
      MU = 16                                                          ,162
      IF(ZAV-1.E+5)4200,8000,8000
C
C     THIS IS WING WITHOUT BODY.  FOR HEIGHT OF FLAT WING
C     USE AVERAGE OF Z FOR 1ST POINT OF 1ST PERCENT LINE AND
C     1ST PT. OF LAST LINE.                                            ,163
 4000 CONTINUE                                                         ,164
      MU = 18                                                          ,165
      JZAV=W(3)                                                        ,166
      ZAVZER=W(JZAV+2)                                                 ,167
      KZAV=W(3*NWPCL)                                                  ,168
      ZAVHUN=W(KZAV+2)                                                 ,169
      ZAV=.5*(ZAVZER+ZAVHUN)
C
C     THE FLAT WING WILL HAVE Z=ZAV FOR ALL POINTS                     ,170
 4200 CONTINUE                                                         ,171
      MU = 20                                                          ,172
      J=3*NWPCL+3                                                      ,173
      DO 4220 I=J,NWEL,3                                               ,174
 4220 W(I)=ZAV
C
C     WRITE FLAT, TRANSFORMED WING ON OUTPUT TAPE                      ,175  ,176
      CALL TFLATW(W,NWPCL,WTITL,DAT,LD)
C
C     WRITE FLAT WING PERCENT LINES ON LD                              ,177
      WRITE(LD) (W(I),I=1,NWEL)                              LD 4
C                                                                      ,178  ,179  ,180  ,181  ,182
      IF(BODY) GO TO 4500
C     WING ALONE.  WRITE DUMMY RECORDS ON LD FOR WING-BODY
C     INTERSECTION POINTS.                                             ,183  ,184  ,185
```

```
      MU = 22                                                             ,186
      JREC(1)=5                                                           ,187
      WRITE(LD)JREC                                       LD 5            ,188  ,189  ,190
      WRITE(LD)DUM                                        LD 6            ,191  ,192  ,193
      IOK=0                                                               ,194
      GO TO 5000
C
C     FIND INTERSECTIONS OF FLAT WING WITH BODY                           ,195
 4500 CONTINUE                                                            ,196
      MU = 24                                                             ,197
      CALL WBXX(B,NBPCT,W,NWPCL,1.E-4,KODEBS,S,BW,NU)                     ,198
      IF(NU)8000,4700,8000
C
C     WRITE WING-BODY INTERSECTION POINTS ON OUTPUT TAPE                  ,199
 4700 CALL TFLATX(LO,DAT,NWPCL,W,BW,NU)                                   ,200
      MU = 26                                                             ,201
      IF(NU)8000,4800,8000
C
C     WRITE INTERSECTICN POINTS ON LD                                     ,202
 4800 MU = 28                                                             ,203
      IREC(1)=5                                                           ,204
      IREC(2)=3*NWPCL                                                     ,205
      IREC(3)=0                                                           ,206
      WRITE(LD)IREC                                       LD 5            ,207  ,208  ,209
      WRITE(LD) ((BW(I,J),I=1,3),J=1,NWPCL)               LD 6
C                                                                         ,210  ,211  ,212  ,213  ,214
                                                                          ,216  ,217
 5000 CONTINUE                                                            ,218
      END FILE LD                                                         ,219
      REWIND LD
C                                                                         ,220
 6000 IF(.NOT. BODY)GO TO 9000
C
C     WRITE TAPE LB FOR AERO AND PANEL LINKS.                             ,221  ,222  ,223
      REWIND LB
C     RECORD 1 = XYZ CCMPONENTS OF UNIT VECTOR ALONG X-PRIME AXIS,
C           FOLLOWED BY ZAV = Z-COORD. OF FLAT WING,
C           FOLLOWED BY TRANSFORMATION CONSTANTS TO CONVERT OLD
C           STATIONS TO NEW STATIONS -
C               NEW X=ALFBET(1)*(OLD X)+ALFBET(2)
C                                                                         ,224
      WRITE(LB)(RT(I,1),I=1,3),ZAV,ALFBET                 LB 1
C
C     PREPARE TO FIND POINTS ON CURVE OF BODY RADIUS VS. X,
C     AND CURVE OF Z-COORD. OF CENTROID VS. X.
C     THERE ARE NX POINTS.  IF THERE IS NO WING,THEN
C     THE PTS. ARE EQUALLY SPACED IN X.  IF THERE IS A WING
C     THEN THE POINTS WILL BE EQUALLY SPACED FROM NOSE TO
C     INTERSECTION OF LEADING EDGE WITH BODY, AND EQUALLY
C     SPACED FROM THERE ON AFT.  THESE TWO SPACINGS
C     WILL 3E APPROX. EQUAL.
C                                                                         ,225  ,226  ,227  ,228  ,229

      MU = 30                                                             ,231
```

```
      BLENG=STA(NSTA)                                                    ,232
      IF(WING)GO TO 6100                                                 ,233  ,234  ,235
C
      XNRX=NRX-1                                                         ,236
      DO 6050 I=1,NRX                                                    ,237
      XIM=I-1                                                            ,238
 6050 RX(1,I)=(XIM/XNRX)*BLENG                                           ,239  ,240
      INDEX=NRX                                                          ,241
      GO TO 6150
C                                                                        ,242
 6100 BLN=BW(1,1)                                                        ,243
      BLR=BLENG-BLN                                                      ,244
      NRXN=(BLN/BLENG)*FLOAT(NRX-1)+1.5                                  ,245
      NRXR=NRX-NRXN+1                                                    ,246
      XNRX=NRXN-1                                                        ,247
      DO 6110 I=1,NRXN                                                   ,248
      XIM=I-1                                                            ,249
 6110 RX(1,I)=(XIM/XNRX)*BLN                                             ,250  ,251
      XNRX=NRXR-1                                                        ,252
      DO 6120 I=2,NRXR                                                   ,253
      XIM=I-1                                                            ,254
      J=NRXN+I-1                                                         ,255
 6120 RX(1,J)=BLN+(XIM/XNRX)*BLR                                         ,256  ,257
      INDEX=NRXN
C                                                                        ,258
 6150 CONTINUE
C
C     THE X VALUES ARE NOW IN THE RX ARRAY.
C     THE RADIUS FOR EACH GIVEN STATION STARTS IN RZ(1,1).
C     THE Z-COORD. OF CENTROID FOR EACH STA STARTS IN RZ(1,2)
C     INTERPOLATE TO GET R,Z CORRESPONDING TO X-VALUES IN RX.
C                                                                        ,259
      MU = 32                                                            ,260
      DO 6300 I=1,NRX
C                                                                        ,261
      DO 6250 J=1,2                                                      ,262
      CALL BITURP(STA,RZ(1,J),1,NSTA,KCDEBR,RX(1,I),RX(J+1,I),NU)        ,263
      IF(NU)8000,6250,6200                                               ,264
 6200 IF(NU-2)6250,6250,8000                                             ,265
 6250 CONTINUE
C                                                                        ,266  ,267
 6300 CONTINUE
C     WRITE RECORDS 2,3                                                  ,268  ,269
      WRITE(LB)NRX,INDEX                                       LB 2      ,270  ,271  ,272
      WRITE(LB) ((RX(I,J),I=1,3),J=1,NRX)                      LB 3      ,273  ,274  ,275  ,276  ,277  ,277
                                                                         ,279  ,280
      END FILE LB                                                        ,281
      REWIND LB
C                                                                        ,282
      WRITE(LO,6400) DAT,(J,(RX(I,J),I=1,3),J=1,NRX)                     ,283  ,284  ,285  ,286  ,287  ,287
                                                                         ,289  ,290  ,291  ,292
 6400 FORMAT(1H1 53X 2A6/38H BODY RADIUS AND Z-COORDINATE OF BODY
     ,23HCENTROID VERSUS X-PRIME/5H0 NO. 10X 7HX-PRIME 18X
     ,6HRADIUS 13X 7HZ-PRIME /(1H I3,F18.4,F24.4,F20.4))
```

```
C
C
      IOK=0                                                          ,293
      IF(BODY .AND. .NCT. WING) NBODY=-1                             ,294  ,295  ,296
      GO TO 9000
C
C     ERROR.  SKIP OVER DATA CARDS TU NEXT CASE, OR
C     UNTIL   END OF DATA   CARD IS FOUND                            ,297
 8000 CONTINUE                                                       ,298
      IOK=MU                                                         ,299
      IF(MU-2)8100,810C,8005                                         ,300
 8005 WRITE(LO,8010)MU,NU(1),(IREC(I),I=1,4)                         ,301  ,302  ,303  ,304  ,305  ,306

 8010 FORMAT(/12HOTFLAT ERROR I3,7H,  CODE I3,
     1 /7H IREC = 4I10)
      IF(MU .LT. 14  .CR.  MU .GT. 18) GO TO 8100                    ,307  ,308  ,309
      WRITE(LO,8020)                                                 ,310  ,311
 8020 FORMAT(45HOTRANSFORMED WING (BEFORE FLATTENING) FOLLOWS )
      CALL TFLATW(W,NWPCL,WTITL,DAT,LO)                              ,312
C
 8100 READ(LI,8110)KIND                                              ,313  ,314  ,315
 8110 FORMAT(A6)
      DO 8200 I=1,3                                                  ,316
      IF(KIND-NAME(I))8200,8300,8200                                 ,317
 8200 CONTINUE                                                       ,318  ,319
      GO TO 8100
C                                                                    ,320
 8300 BACKSPACE LI
C                                                                    ,321
 9000 CONTINUE                                                       ,322
      DO 9100 J=1,6                                                  ,323
      DO 9100 I=1,3                                                  ,324
 9100 LGDEF(I,J) = .FALSE.                                           ,325  ,326  ,327
      RETURN                                                         ,328
      END                                                            ,329
```

```
      SUBROUTINE TFLAT1(CHD,NSTA,STA,NBPCT,B,S,E,RT,ALFBET,RZ,EPRC,NU)
C
      DIMENSION RZ(51,2),STA(51),B(2448),RT(3,4),ALFBET(2),NU(3),S(100,2
     1), E(2000),EP(5),BEND(3,2),AREA(2),RAD(2),UNITZ(3)
C
C     USED BY TFLAT TO TRANSFORM BODY MERIDIAN LINES TO A
C       NEW COORDINATE SYSTEM WITH ORIGIN AT THE CENTROID OF
C       THE FIRST BODY SECTION.  THE NEW +X AXIS PASSES THRU
C       THE CENTROID OF THE LAST BODY SECTION.
C       IT IS ASSUMED THAT BODY MERIDIAN LINES ARE GIVEN FOR A HALF-BODY
C         AND THAT THE BODY IS SYMMETRICAL ABOUT A PLANE PARALLEL TO THE
C         XZ PLANE.
C     TRANSFORMED BODY MERIDIAN LINES ARE SCALED (INDIVIDUALLY) SO
C       THAT 1ST PT IN EACH LINE HAS X=0 AND EACH LINE HAS SAME
C       X-LENGTH (LENGTH BETWEEN END CENTROIDS).
C INPUTS ARE
C     CHD = CHORD-HEIGHT TOLERANCE.  TO FIND THE CENTROID
C       OF A BODY SECTION, WE CUT EACH MERIDIAN LINE WITH THE
C       X-VALUE OF THE SECTION.  THIS GIVES US A POLYGON OF THE
C       SECTION IN 2-DIMENSIONS (Y,Z).  IF CHD .GT. 0, WE THEN
C       INTERPOLATE ADDITIONAL POINTS ON THE POLYGON TO BETTER
C       APPROXIMATE A SMOOTH BODY.  CHD IS USED BY S/R ENRYCH.
C     NSTA = NO. OF STATIONS
C     NBPCT = NO. OF BODY MERIDIAN LINES
C INPUT-OUTPUTS ARE
C     STA = ARRAY OF BODY STATIONS.  ON OUTPUT, SCALED TO AGREE
C       WITH TRANSFORMED BODY.
C     B = ARRAY OF BODY MERIDIAN LINES (WITH HEADER).
C     EPRC IS USED BY BODCR TO SET Z-CENTROID AND RADIUS =0 IF .LE. EPRC.
C OUTPUTS ARE
C     RT = ROTATION, TRANSLATION MATRIX OF THE TRANSFORMATION
C       FROM ORIGINAL SYSTEM TO NEW SYSTEM.
C       COL. 1 IS UNIT VECTOR ALONG NEW +X AXIS
C       COL. 2 IS UNIT VECTOR ALONG NEW +Y AXIS
C       COL. 3 IS UNIT VECTOR ALONG NEW +Z AXIS
C       COL. 4 IS ORIGIN OF NEW XYZ SYSTEM
C              ALL WITH RESPECT TO ORIGINAL XYZ SYSTEM
C     ALFBET = PAIR OF CONSTANTS USED TO RELATE OLD STATION
C       VALUES TO NEW.
C       NEW X = ALFBET(1)*(OLD X)+ ALFBET(2)
C     RZ = ARRAYS OF BODY RADIUS AND Z-COORD. OF CENTROID AT
C       EACH BODY STATION
C SCRATCH ARRAYS ARE  S, E
C
      DATA EP/5*.005/
      EP(5)=ABS(EPRC)
C
C     BEND (BODY ENDS) ARRAY WILL BE COORD. OF CENTROIDS
C     OF THE END SECTIONS OF THE BODY                               ,1
 100  BEND(1,1)=STA(1)                                              ,2
      BEND(1,2)=STA(NSTA)
C
C     FIND EQUIV. RADIUS AND CENTROID OF BODY END SECTIONS          ,3
```

```
       TFLAT1                                                           07/31/65
          EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT     -    INTERNAL FORMULA NUMBER(S)

       DO 200 I=1,2                                                        ,4
       MU=I                                                                ,5
       CALL BODCR(B,NBPCT,BEND(1,I),EP,CHD,2000,S(1,I),E,NE,
      , BEND(2,I),BEND(3,I),AREA(I),RAD(I),NU)                             ,6
       IF(NU)8000,200,8000                                                 ,7
 200   CONTINUE
C
C      FIND TRANSFORMATION MATRICES TO NEW SYSTEM                          ,8    ,9
 300   BLENG=UVECN(BEND,BEND(1,2),RT,0,3)                                  ,10
       DATA UNITZ/0.,0.,1./
       IZXP=1                                                              ,11
       CALL VCROS(UNITZ,RT,RT(1,2),DZXP,IZXP)                              ,12
       IXPYP=1                                                             ,13
       CALL VCROS(RT,RT(1,2),RT(1,3),DXPYP,IXPYP)                          ,14
       MU=3                                                                ,15
       IF(BLENG .EQ. 0. .OR. IZXP .NE. 0 .OR. IXPYP .NE. 0) GO TO 8000     ,16   ,17   ,18
       CALL TRAV(RT(1,4),BEND,3)
C      RT = TRANSFORMATION ARRAY   IS COMPLETE.
C      BLENG = LENGTH OF BODY ALONG NEW X AXIS
C
C      TRANSFORM BODY MERIDIAN LINES TO NEW SYSTEM                         ,19
 500   CALL TRAPCT(B,NBPCT,RT)
C
C      NOW SCALE SO ALL MERIDIAN LINES WILL START AT
C        X=0, END AT X=BLENG.                                              ,20
       MU=4                                                                ,21
 600   CALL BSCALE(B,NBPCT,BLENG,NU)                                       ,22
       IF(NU)8000,1000,8000
C
C      SCALE BODY STA ARRAY TO CORRESPOND TO NEW COORD. SYSTEM             ,23
 1000  MU=5                                                                ,24
       D=STA(NSTA)-STA(1)                                                  ,25
       IF(D)1200,8000,1200                                                 ,26
 1200  ALFBET(1)=BLENG/D                                                   ,27
       ALFBET(2)=-ALFBET*STA(1)                                            ,28
       DO 1250 I=1,NSTA                                                    ,29
 1250  STA(I)=ALFBET*STA(I)+ALFBET(2)
C
C      FIND RADIUS AND Z-COORD. OF CENTROID AT EACH STATION.               ,30   ,31
 2000  MU=6                                                                ,32
       DO 2100 I=1,NSTA                                                    ,33
       CALL BODCR(B,NBPCT,STA(I),EP,CHD,2000,S,E,NE,
      , Y,RZ(I,2),A,RZ(I,1),NU)                                            ,34
       IF(NU)8000,2100,8000                                                ,35
 2100  CONTINUE
C                                                                          ,36   ,37
       NU=0                                                                ,38
       GO TO 9000
C
C      ERROR                                                               ,39
 8000  CONTINUE                                                            ,40
       LTAPE=NU(2)                                                         ,41
       IF(LTAPE .LE. 0)GO TO 8500                                          ,42   ,43   ,44
       WRITE(LTAPE,8100)MU,NU(1)                                           ,45   ,46   ,47
```

```
          TELATI                                                                    07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

 8100 FORMAT(/13H0TELATI ERROR I3,7H,  CODE I3,5X
     1 49HFAILURE IN TRANSFORMING BODY MERIDIAN LINES, ETC.)
      IF(MU .NE. 6)GO TO 8500                                                 ,48   ,49   ,50
      WRITE(LTAPE,8200)STA(I)                                                 ,51   ,52   ,53
 8200 FORMAT(8H STATION F10.4)
 8500 NU=MU
C                                                                             ,54
 9000 RETURN                                                                  ,55
      END                                                                     ,56
```

```
C      SAME AS BODYIM EXCEPT DOES NOT IDENTIFY POINTS WHICH LIE
C      IN A BODY DEFINING SECTION.
C
C      USED BY TFLAT TO PRINT BODY MERIDIAN LINES.
C      ALSO COMPUTES RHO, THETA FOR EACH POINT.
C      THE RHO, THETA ORIGIN IS ON BODY AXIS.
C
       SUBROUTINE TFLATM(B,NB,AXIS,TITLE,DAYT,LO,STA)
C
       DIMENSION B(1),AXIS(2),TITLE(12),DAYT(2),STA(1),XYZ(3)
       LOGICAL NEWPCT,WTITLE
       DATA MAXLIN,DEGRAD/45,O206712273406/
C
C      B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C      STA IS ARRAY OF BODY STATIONS (NOT USED BY TFLATM)
C      MAXLIN = MAX NO. OF LINES PER PAGE
C      MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C        ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
       MAXEL=3*(MAXLIN-2)                                         ,1
       NELREM=0
C                                                                 ,2
       DO 4000 K=1,NB
C      WORKING ON KTH MERIDIAN LINE                               ,3
       JK=3*K                                                     ,4
       NPTS=B(JK-1)                                               ,5
       LOCST=B(JK)                                                ,6
       LOCEND=LOCST+3*NPTS-1
C      THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C        IN B(LOCST).
C      THE Z OF THE LAST POINT IS IN B(LOCEND).
C      FOR PAGE CONTROL WE WRITE A BLOCK OF POINTS FROM
C        B(LA) TO B(LB).                                          ,7
       LA=LOCST                                                   ,8
       NEWPCT=.TRUE.                                              ,9
       ISTA=1
C
C      WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN   ,10
   200 NELREM=NELREM-9
C      THIS ALLOWS FOR HEADING                                    ,11
       IF(NELREM-6)300,400,400
C      NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE       ,12
   300 WTITLE=.TRUE.                                              ,13
       NELREM=MAXEL-6                                             ,14
       GO TO 500                                                  ,15
   400 WTITLE=.FALSE.
C                                                                 ,16
   500 LB=MINO (LOCEND,LA+NELREM-1)                               ,17
       NL=(LB-LA+1)/3                                             ,18
       IF(.NOT. WTITLE) GO TO 600                                 ,19    ,20    ,21
       WRITE(LO,550)TITLE,DAYT                                    ,22    ,23    ,24
   550 FORMAT(1H1 12A6/54X 2A6)
   600 IF(NEWPCT) GO TO 700
```

```
          TFLATM                                                       09/14/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C       WRITE HEADING                                                   ,25   ,26   ,27
        WRITE(LO,650)K                                                  ,28   ,29   ,30
  650   FORMAT(19H0BODY MERIDIAN LINE I4,14H    (CONTINUED))
        GO TO 800                                                       ,31
  700   WRITE(LO,750)K                                                  ,32   ,33   ,34
  750   FORMAT(19H0BODY MERIDIAN LINE I4)
        NUM=0                                                           ,35
  800   WRITE(LO,850)                                                   ,36   ,37
  850   FORMAT(14X1HX10X1HY10X1HZ12X3HRHO7X5HTHETA)
C
C       READY TO WRITE PCINTS, COMPUTING RHO-THETA AS WE GO
        DO 2000 L=1,NL                                                  ,38
        NUM=NUM+1                                                       ,39
        CALL TRAV(XYZ,B(LA),3)                                          ,40
        DY=XYZ(2)-AXIS(1)                                               ,41
        DZ=XYZ(3)-AXIS(2)                                               ,42
        RHO=SQRT(DY*DY+DZ*DZ)                                           ,43
        IF(RHO)1100,1100,1200                                           ,44
 1100   THETA=0.                                                        ,45
        GO TO 1300                                                      ,46
 1200   THETA = ATN1(ABS(DY),DZ)*DEGRAD
C                                                                       ,47
 1300   CONTINUE
C                                                                       ,48
 1500   WRITE(LO,1550)NUM,XYZ,RHO,THETA                                 ,49   ,50   ,51
 1550   FORMAT(I4,4X,3F11.4,3X,2F11.4)
        LA=LA+3                                                         ,52
 2000   CONTINUE
C                                                                       ,53   ,54
        NELREM=NELREM-3*NL                                              ,55
        IF(LB-LOCEND)3000,4000,4000                                     ,56
 3000   NEWPCT=.FALSE.                                                  ,57
        LA=LB+1                                                         ,58
        GO TO 200
C                                                                       ,59
 4000   CONTINUE
C                                                                       ,60   ,61
        RETURN                                                          ,62
        END                                                             ,63
```

```
        TFLATW                                                             09/14/65
     EXTERNAL FURMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TFLATW(B,NB,TITLE,DAYT,LO)

C     USED BY TFLAT TO PRINT WING PERCENT LINES.
C       (ADAPTED FROM BODY1M)

      DIMENSION B(1),TITLE(12),DAYT(2),XYZ(3)
      LOGICAL NEWPCT,WTITLE
      DATA MAXLIN/45/
C
C     B IS ARRAY OF NB MERIDIAN LINES, HEADED BY INDEX.
C     MAXLIN = MAX NO. OF LINES PER PAGE
C     MAXEL IS MAX NO. OF XYZ ELEMENTS THAT WILL GO ON A PAGE,
C       ASSUMING NO HEADINGS.  WE ALLOW FOR HEADINGS LATER.
C
      MAXEL=3*(MAXLIN-2)                                                  ,1
      NELREM=0
C                                                                         ,2
      DO 4000 K=1,NB
C     WORKING ON KTH MERIDIAN LINE                                        ,3
      JK=3*K                                                              ,4
      PCT=B(JK-2)                                                         ,5
      NPTS=B(JK-1)                                                        ,6
      LOCST=B(JK)                                                         ,7
      LOCEND=LOCST+3*NPTS-1
C     THE X OF THE FIRST POINT IN THE MERIDIAN LINE IS
C       IN B(LOCST).
C     THE Z OF THE LAST POINT IS IN B(LOCEND).
C     FOR PAGE CONTROL WE WRITE A BLOCK OF POINTS FROM
C       B(LA) TO B(LB).                                                   ,8
      LA=LOCST                                                            ,9
      NEWPCT=.TRUE.                                                       ,10
      ISTA=1
C
C     WE KEEP COMING BACK TO HERE UNTIL ALL POINTS ARE WRITTEN            ,11
  200 NELREM=NELREM-9
C     THIS ALLOWS FOR HEADING                                             ,12
      IF(NELREM-6)300,400,400
C     NOT ENOUGH ROOM FOR 2 POINTS, SO WILL START NEW PAGE                ,13
  300 WTITLE=.TRUE.                                                       ,14
      NELREM=MAXEL-6                                                      ,15
      GO TO 500                                                           ,16
  400 WTITLE=.FALSE.
C                                                                         ,17
  500 LB=MINO (LOCEND,LA+NELREM-1)                                        ,18
      NL=(LB-LA+1)/3                                                      ,19
      IF(.NOT. WTITLE) GO TO 600                                          ,20    ,21    ,22
      WRITE(LO,550)TITLE,DAYT                                             ,23    ,24    ,25
  550 FORMAT(1H1 12A6/54X 2A6)
  600 IF(NEWPCT) GO TO 700
C     WRITE HEADING                                                       ,26    ,27    ,28
      WRITE(LO,650) K,PCT                                                 ,29    ,30    ,31
  650 FORMAT(24HOWING PERCENT CHORD LINE I3,1H,F9.4, 14H   (CONTINUED))
      GO TO 800                                                           ,32
```

```
         IFLATW                                                             09/14/65
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

  700 WRITE(LO,750) K,PCT                                                 ,33   ,34   ,35
  750 FORMAT(24HOWING PERCENT CHORD LINE I3,1H,F9.4)
      NUM=0                                                               ,36
  800 WRITE(LO,850)                                                       ,37   ,38
  850 FORMAT(14X 1HX 1CX 1HY 10X 1HZ)
C
C     READY TO WRITE PCINTS
      DO 2000 L=1,NL                                                      ,39
      NUM=NUM+1                                                           ,40
      CALL TRAV(XYZ,B(LA),3)
C                                                                         ,41
      WRITE(LO,1550)NUM,XYZ                                               ,42   ,43   ,44
 1550 FORMAT(1H I3,4X,3F11.4)
      LA=LA+3
C                                                                         ,45
 2000 CONTINUE
C                                                                         ,46   ,47
      NELREM=NELREM-3*NL                                                  ,48
      IF(LB-LOCEND)300C,4000,4000                                         ,49
 3000 NEWPCT=.FALSE.                                                      ,50
      LA=LB+1                                                             ,51
      GO TO 200
C                                                                         ,52
 4000 CONTINUE
C                                                                         ,53   ,54
      RETURN                                                              ,55
      END                                                                 ,56
```

```
         TFLATX                                                                       09/14/65
      EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TFLATX(LO,DAT,NW,W,WB,NU)
C
C     USED BY TFLAT TO PRINT WING-BODY INTERSECTIONS.
C     ADAPTED FROM PART OF WBXUL
C     W = ARRAY OF WING PERCENT LINES (WITH HEADER)
C     WB = ARRAY OF WING-BODY INTERSECTION POINTS
C     NW = NO OF WING PERCENT LINES
C
      DIMENSION DAT(2),W(1),WB(1),PT(3),BAXIS(2)
C
      DATA BAXIS/0.,0./,DEGRAD/0206712273406/
C
C
      NU=0                                                              ,1
      WRITE(LO,100)DAT,BAXIS                                            ,2     ,3     ,4
  100 FORMAT(1H1 53X 2A6/23H INTERSECTIONS OF FLAT
     , 42HWING PERCENT CHORD LINES WITH BODY SURFACE /
     , 41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( F9.4,1H,F9.4,1H)/
     , 21HO(PRIMED-AXIS SYSTEM)/
     , 13HONO.  PERCENT 6X 1HX 9X 1HY 9X 1HZ 10X 3HRHO 6X 5HTHETA//)
C
      IPCT=1                                                            ,5
      IWBX=1
C                                                                       ,6
      DO 1000 J=1,NW                                                    ,7
      PCT=W(IPCT)                                                       ,8
      IPCT=IPCT+3                                                       ,9
      CALL TRAV(PT,WB(IWBX),3)                                          ,10
      IWBX=IWBX+3                                                       ,11
      IF(PT .GT. 1.E+29) GO TO 900                                      ,12    ,13    ,14
      DY=PT(2)                                                          ,15
      DZ=PT(3)                                                          ,16
      RHO=SQRT(DY*DY+DZ*DZ)                                             ,17
      IF(RHO)820,820,830                                                ,18
  820 THETA=0.                                                          ,19
      GO TO 850                                                         ,20
  830 THETA=ATN1(ABS(DY),DZ)*DEGRAD                                     ,21
  850 WRITE(LO,860)J,PCT,PT,RHO,THETA                                   ,22    ,23    ,24
  860 FORMAT(1H I2,F10.4,1X 3F10.4,2X 2F10.4)
      GO TO 1000                                                        ,25
  900 WRITE(LO,910)J,PCT                                                ,26    ,27    ,28
  910 FORMAT(1H I2,F10.4,6X 15HNO INTERSECTION)
      NU=1                                                              ,29
 1000 CONTINUE
C                                                                       ,30    ,31
      RETURN                                                            ,32
      END                                                               ,33
```

```
          THETAB                                                                09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

       SUBROUTINE THETAB
       COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
      1NWRITE,NBODY,NWING,XMACH,SYM,KACE
       COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
      1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
      2,NPLANE,NPLN1
       COMMON /COM2/ NPLNB,NPLNW,JLEAD,JTRAIL,IMID,NPTS(16),X(16,90),Y(16
      1,90),Z(16,90),XCEPT(21),XCEPTB(21),XCEPTW(16),YCEPTW(16),ZCEPTW(16
      2),CODEBW(16),KPANEL(15,20),XCOR(16,21),YCOR(16,21),ZCOR(16,21),XIN
      3T(15,20,2),YINT(15,20,2),ZINT(15,20,2),XCEN(15,20),YCEN(15,20),ZCE
      4N(15,20),XCON(15,20),YCON(15,20),ZCON(15,20),AREA(15,20),ARAT(15,2
      50),THETA(15,20),ALPHA(15,20),CHORD(15,20)
       DIMENSION C(4),P(3,5),Q(3,5),LOC(3)

C      SUBROUTINE TO CALCULATE BODY PANEL
C      THETA ANGLES

C      DEFINE FUNCTIONS TO BE USED IN CALCULATIONS
C      DETR CALCULATES MATRIX DETERMINANT
C      ANGL CALCULATES THETA-INCLINATION ANGLE USING
C      ARCCOSINE FUNCTION ROUTINE
       DETR(A1,A2,A3,B1,B2,B3)=A1*(B2-B3)-A2*(B1-B3)+
      1A3*(B1-B2)
       ANGL(C1,C2)=ACOS(C1/SQRT(C1*C1+C2*C2))

C      DEFINE CONSTANTS
       E=0.001                                                            ,1
       F=0.000001

C      CALCULATE PANEL THETA ANGLES                                       ,2
 50    DO 70 I=1,NPER1                                                    ,3
       DO 65 J=1,NPLN1                                                    ,4
       IF (ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))-F) 55,55,60                     ,5
 55    THETA(I,J)=0.                                                      ,6
       GO TO 65                                                           ,7
 60    C1= DETR(XCOR(I,J),XCOR(I,J+1),XCOR(I+1,J+1),
      1YCOR(I,J),YCOR(I,J+1),YCOR(I+1,J+1))                               ,8
       C2=-DETR(XCOR(I,J),XCOR(I,J+1),XCOR(I+1,J+1),
      1ZCOR(I,J),ZCOR(I,J+1),ZCOR(I+1,J+1))                               ,9
       THETA(I,J)=(ZCOR(I+1,J+1)-ZCOR(I,J+1))/
      1ABS(ZCOR(I+1,J+1)-ZCOR(I,J+1))*ANGL(C1,C2)                         ,10
 65    CONTINUE                                                           ,11    ,12
 70    CONTINUE                                                           ,13    ,14
 100   RETURN                                                             ,15
       END                                                                ,16
```

```
         TRAPCT                                                        07/31/65
       EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TRAPCT(B,N,RT)
      DIMENSION B(3,1),RT(3,4)
C
C     GIVEN AN ARRAY B OF N BODY MERIDIAN LINES OR WING PERCENT LINES,
C     (WITH HEADER WHICH DESCRIBES EACH PERCENT LINE), APPLY THE
C     TRANSFORMATION RT TO EACH POINT.
C
C     1ST POINT (X,Y,Z) STARTS IN B(1,N+1).
      J=N+1                                                              ,1
C     NPT = TOTAL NO. OF POINTS IN B.
      NPT=0                                                              ,2
      DO 100 I=1,N                                                       ,3
      NP=B(2,I)                                                          ,4
  100 NPT=NPT+NP
C     TRANSFORM TO NEW COORD. SYSTEM                                     ,5    ,6
      DO 200 I=1,NPT                                                     ,7
      CALL TRCTPT(RT,RT(1,4),2,B(1,J),B(1,J))                            ,8
  200 J=J+1
C                                                                        ,9    ,10
      RETURN                                                             ,11
      END                                                                ,12
```

```
          TRAV                                                        07/31/65
     EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TRAV(A,B,N)
      DIMENSION A(1),B(1)
C
      DO 100 I=1,N                                                          ,1
  100 A(I)=B(I)
C                                                                           ,2       ,3
      RETURN                                                                ,4
      END                                                                   ,5
```

```
     TROTPT                                                          07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE TROTPT(R,T,N,P,Q)
C
      DIMENSION R(3,3),T(3),P(3),Q(3),A(3)
C
C     TRANSLATION AND ROTATION OF A POINT
C     P IS A POINT (X,Y,Z) IN FIRST SYSTEM, Q IN 2ND
C     N = 1 IF Q GIVEN, P WANTED.  N = 2 IF P GIVEN, Q WANTED
C     R AND T ARE ROTATION, TRANSL. MATRICES TO MAP Q INTO P
C     P = R Q +T
C
      IF (N-1)20, 1,20                                        ,1
    1 DO 5 I=1,3                                              ,2
    5 A(I)=Q(I)                                               ,3     ,4
   10 DO 15 I=1,3                                             ,5
      U=0.                                                    ,6
      DO 12 J=1,3                                             ,7
   12 U=U+R(I,J)*A(J)                                         ,8     ,9
   15 P(I)=U+T(I)                                             ,10    ,11
      GO TO 50                                                ,12
   20 DO 25 I=1,3                                             ,13
   25 A(I)=P(I)-T(I)                                          ,14    ,15
      DO 40 I=1,3                                             ,16
      U=0.                                                    ,17
      DO 30 J=1,3                                             ,18
   30 U=U+R(J,I)*A(J)                                         ,19    ,20
   40 Q(I)=U                                                  ,21    ,22
   50 RETURN                                                  ,23
      END                                                     ,24
```

05/22/67

TTAPE - EFN SOURCE STATEMENT - IFN(S) -

```
      SUBROUTINE TTAPE(KIND,IN,JN,NTIN,NTOUT,A,B,C)
      DIMENSION A(1),B(1),C(1)
      IF(KIND)150,50,150
   50 CONTINUE
      DO 100 J=1,JN
      READ(NTIN)(A(I),I=1,IN)                                         6
  100 WRITE(NTOUT)(A(I),I=1,IN)                                       11
      GO TO 250
  150 CONTINUE
      DO 200 J=1,JN
      READ(NTIN)(A(I),B(I),C(I),I=1,IN)                               22
  200 WRITE(NTOUT)(A(I),B(I),C(I),I=1,IN)                             29
  250 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TVEL(A,B,C,D,ALPHAT,UBWT,VBWT,WBWT,UWWT,VWWT,WWWT,AN,
     $CHORD,THICK,NROW)
C     COMPUTE VELOCITY COMPONENTS INDUCED BY WING SOURCES
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION A(1),B(1),C(1),D(1),ALPHAT(1),UBWT(1),VBWT(1),WBWT(1),
     1UWWT(1),VWWT(1),WWWT(1),AN(1),NROW(1),CHORD(1)
      DIMENSION ATLE(20),TSL(2)
      DATA TSL/5HGIVEN,5HGIVEN/
      NS=NBODY+1
      NPANEL=NBODY+NWING
      IF (NBODY .EQ. 0) GO TO 95
      DO 90 J=1,NBODY
      UBWT(J)=0.
      VBWT(J)=0.
      WBWT(J)=0.
      AN(J)=0.
   90 CONTINUE
   95 DO 100 J=1,NWING
      UWWT(J)=0.
      VWWT(J)=0.
      WWWT(J)=0.
  100 CONTINUE
      GO TO (110,500,120),KACE
  110 NC=NWING/NROW(1)
      NR=NROW(1)
      GO TO 130
  120 NC=NWING/NROW(2)
      NR=NROW(2)
  130 N=0
      NWINC=NWING+NC
      IGO=INTURP(TSL,2,NTAPEI,NTAPEO)                                  36
      IGO=IGO+1
      GO TO (132,136,136),IGO
C     INPUT THICKNESS SLOPES
  132 READ (NTAPEI,6000) (A(I),I=1,NWINC)                              39
      IJ=0
      NR1=NR+1
      DO 134 J=1,NC
      J1=(J-1)*(NR+1)
      ATLE(J)=A( J1+1)
      DO 133 I=2,NR1
      IJ=IJ+1
      IJ1=I+J1
      ALPHAT(IJ)=A(IJ1)
```

```
   TVEL        -  EFN   SOURCE STATEMENT  -  IFN(S)  -                  05/22/67

  133 CONTINUE
  134 CONTINUE
      GO TO 140

C     USE PREVIOUSLY CALCULATED PANEL SLOPES
C     AND COMPUTE LEADING EDGE SLOPES
  136 CALL FSF(1,NTAPEC,IRR)                                                67
      READ (NTAPEC) (ALPHAT(I),I=1,NWING)                                   68
      REWIND NTAPEC                                                         75
      NR1=NR-1
      DO 138 J=1,NC
      J1=(J-1)*NR
      JN=NBODY+J1
      AT=-ALPHAT(J1+1)
      DO 137 I=1,NR1
      IJ=J1+I
      IJN=NBODY+IJ
      AT=AT-(ALPHAT(IJ)+ALPHAT(IJ+1))*CHORD(IJN+1)/
     -CHORD(JN+1)
  137 CONTINUE
      ATLE(J)=AT
  138 CONTINUE
  140 CONTINUE
      CALL FSF(8,NTAPEC,IRR)                                                98
      WRITE (NTAPEC) NWINC                                                  99
      DO 200J=1,NWINC
      READ (NTAPEA) (D(I),I=1,NPANEL)                                      102
      READ (NTAPEB) (A(I),B(I),C(I),I=1,NPANEL)                            107
      JT=J-N*(NR+1)
      IF (JT) 160,160,150
  150 N=N+1
      WT=ATLE(N)
      GO TO 170
  160 JN=J-N
      WT=ALPHAT(JN)

  170 CONTINUE
      WRITE (NTAPEC) WT                                                    123
      DO 190 I=1,NPANEL
      IF (I .GT. NBODY) GO TO 180
      UBWT(I)=UBWT(I)+A(I)*WT
      VBWT(I)=VBWT(I)+B(I)*WT
      WBWT(I)=WBWT(I)+C(I)*WT
      AN(I)=AN(I)+D(I)*WT
      GO TO 190
  180 IN=I-NBODY
      UWWT(IN)=UWWT(IN)+A(I)*WT
      VWWT(IN)=VWWT(IN)+B(I)*WT
      WWWT(IN)=WWWT(IN)+C(I)*WT
  190 CONTINUE
  200 CONTINUE
  500 CONTINUE
      END FILE NTAPEC
      REWIND NTAPEC                                                        158
      RETURN                                                               159
 6000 FORMAT (7F10.0)
      END
```

```
  UFOOL                                                          07/31/65
 7094 RELMOD ASSEMBLY.

      $IBLDR UFOOL                                                                      UF000000
 ASSEMBLED TEXT.
      $TEXT  UFOOL                                                                      UF000001

                         *FOOL       MICHAEL SYNGE          6-9247      ASSEMBLED  26 JAN 65
                               ENTRY   FOOL
                          IFOOL ENTRY   FOOL
                         *
BINARY CARD (NOT PUNCHED)
     00000  0020 00 4 00001  10000   FOOL  TRA     1,4
                                    *
                 00000  01111         END
 CONTROL DICTIONARY
      $CDICT UFOOL                                                                      UF000002

BINARY CARD (NOT PUNCHED)
          000001000000              PREFACE     START=0,LENGTH=1,TYPE=7094,CMPLX=5
          000004000005
          642646464360       UFOOL  DECK        LOC=0,LENGTH=1
          000001000000
          254646436060       FOOL   REAL        LOC=0,LENGTH=0
          000000000000
          312646464360       IFOOL  REAL        LOC=0,LENGTH=0
          000000000000

      $DKEND UFOOL                                                                      UF000003

NO MESSAGES FOR THIS ASSEMBLY
          SYMBOL REFERENCE DATA

REFERENCES TO DEFINED SYMBOLS.

  CLASS  SYMBOL   VALUE   REFERENCES

         FOOL     00000
  LCTR    BLCTR
  DUAL    UNDS
  LCTR   //
```

```
      $IBLDR UN07                                                          UN070000
      $FILE  UN07    'UNIT07',C(1),READY,NOLIST,INOUT,BLK=256,HIGH,BIN     UN070001
```

```
 UN08                                           07/31/65
7094 RELMOD ASSEMBLY.

       $IBLDR UN08                                                      UN080000

       $FILE  UN08    (UNIT08),C(2),READY,NOLIST,INOUT,BLK=256,HIGH,BIN UN080001
```

```
 UN09                                                         08/16/65
7094 RELMOD ASSEMBLY.

      $IBLDR UN09                                                          UN090000

      $FILE  UN09    |UNIT09|,A(1),READY,NOLIST,INOUT,BLK=14,HIGH,BCD     UN090001
```

```
   UPDATE                                                           07/31/65
  7094 RELMOD ASSEMBLY.

       $IBLDR UPDATE                                                                UPDA0000
  ASSEMBLED TEXT.
       $TEXT  UPDATE                                                                UPDA0001
                                   *UPDATE    MICHAEL SYNGE         6-9247     ASSEMBLED  21 SEP 64
                                   *     TO UPDATE THE CONTENTS OF AN ARRAY NAME WHEN THE
                                   *     ARRAY IS MOVED (E.G. BY PACKING).
                                   *
                                   *              CALLING SEQUENCE
                                   *
                                   *                  CALL UPDATE(BUF(M-1))
                                   *
                                   *              WHERE M IS THE FIRST WORD OF THE ARRAY
                                   *              (AFTER MOVING) IN THE BUFFER  BUF .
                                   *
                                          ENTRY   UPDATE
                                   *
BINARY CARD (NOT PUNCHED)
      00000  0500 60 4 00003  10000 UPDATE CLA*   3,4
      00001  0621 00 0 00007  10001        STA    NAME
      00002  0500 00 4 00003  10000        CLA    3,4
      00003  0402 00 0 00011  10001        SUB    =1
      00004  0621 00 0 01001  10011        STA    *+1
      00005  0500 00 0 00000  10000        CLA    **
      00006  4320 00 0 00012  10001        ANA    =077777
      00007  0621 00 0 00000  10000  NAME  STA    **
      00010  0020 00 4 00001  10000        TRA    1,4
                                   *
      00011  000000000001     10000        *LORG
      00012  000000077777     10000
                       00000  01111        END
       CONTROL DICTIONARY

          $CDICT UPDATE                                                             UPDA0002

 BINARY CARD (NOT PUNCHED)
          000013000000             PREFACE      START=0,LENGTH=11,TYPE=7094,CMPLX=5
          000004000005
          644724216325         UPDATE DECK      LOC=0,LENGTH=11
          000013000000
          644724216325         UPDATE REAL      LOC=0,LENGTH=0
          000000000000

          $DKEND UPDATE                                                             UPDA0003

 NO MESSAGES FOR THIS ASSEMBLY
```

REFERENCES TO DEFINED SYMBOLS.

| CLASS | SYMBOL | VALUE | REFERENCES |
|---|---|---|---|
| | NAME | 00007 | 1 |
| LCTR | BLCTR | | |
| QUAL | UNQS | | |
| LCTR | // | | |
| | UPDATE | 00000 | |

```
USETAP      -  EFN   SOURCE STATEMENT  -  IFN(S)  -              05/22/67

      SUBROUTINE USETAP
      COMMON DATE(2),NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI
     1,NTAPEO,NBODY,NWING,XMACH,SYM,KACE
      DIMENSION A(210),B(210),C(210)
      DIMENSION DUM(10),NROW(2)
      READ(NTAPEC)NBODY,NWING,XMACH,SYM,KACE                              1
      GO TO ( 50,900, 50),KACE
   50 NPANEL=NBODY+NWING
      READ (NTAPEC) (I,(DUM(J),J=1,10),I=1,NPANEL),NRG,
     1(NROW(I),I=1,NRG),DUMMY                                              7
      NCOLW=NWING/NROW(NRG)
      CALL FSF( 6,NTAPEC,IRR)                                             27
      GO TO (100,900,200),KACE
  100 CONTINUE
      CALL TTAPE(0,NWING,NWING+NCOLW,NTAPEC,NTAPEA,A,B,C)                 30
      END FILE NTAPEA                                                     31
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEA,A,B,C)                       32
      END FILE NTAPEA                                                     33
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEC,NTAPEA,A,B,C)                   34
      END FILE NTAPEA                                                     35
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEC,NTAPEA,A,B,C)                   36
      END FILE NTAPEA                                                     37
      REWIND NTAPEA                                                       38
      CALL TTAPE(1,NWING,NWING+NCOLW,NTAPEC,NTAPEB,A,B,C)                 39
      END FILE NTAPEB                                                     40
      CALL TTAPE(1,NWING,NWING,NTAPEC,NTAPEB,A,B,C)                       41
      END FILE NTAPEB                                                     42
      REWIND NTAPEB                                                       43
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)                       44
      END FILE NTAPEE                                                     45
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)                       46
      END FILE NTAPEE                                                     47
      REWIND NTAPEE                                                       48
      GO TO 900
  200 CONTINUE
      CALL TTAPE(0,NPANEL,NWING +NCOLW,NTAPEC,NTAPEA,A,B,C)               51
      END FILE NTAPEA                                                     52
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEA,A,B,C)                       53
      END FILE NTAPEA                                                     54
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEC,NTAPEA,A,B,C)                   55
      END FILE NTAPEA                                                     56
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEC,NTAPEA,A,B,C)                   57
      END FILE NTAPEA                                                     58
      REWIND NTAPEA                                                       59
      CALL TTAPE(1,NPANEL,NWING +NCOLW,NTAPEC,NTAPEB,A,B,C)               60
      END FILE NTAPEB                                                     61
      REWIND NTAPEB                                                       62
      CALL TTAPE(1,NBODY,NBODY,NTAPEC,NTAPED,A,B,C)                       63
      END FILE NTAPED                                                     64
      CALL TTAPE(1,NWING,NBODY,NTAPEC,NTAPED,A,B,C)                       65
      END FILE NTAPED                                                     66
      CALL TTAPE(1,NBODY,NWING,NTAPEC,NTAPED,A,B,C)                       67
      END FILE NTAPED                                                     68
```

```
      CALL TTAPE(1,NWING,NWING,NTAPEC,NTAPED,A,B,C)
      END FILE NTAPED
      REWIND NTAPED
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      END FILE NTAPEE
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      END FILE NTAPEE
      REWIND NTAPEE
      CALL TTAPE(0,NWING,NBODY,NTAPEC,NTAPEF,A,B,C)
      END FILE NTAPEF
      CALL TTAPE(0,NBODY,NBODY,NTAPEC,NTAPEF,A,B,C)
      CALL TTAPE(0,NBODY,NWING,NTAPEC,NTAPEF,A,B,C)
      END FILE NTAPEF
      REWIND NTAPEF
  900 CONTINUE
      REWIND NTAPEC
      RETURN
      END
```

```
      UVECN                                                              07/31/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      FUNCTION UVECN(A,B,V,K,N)
      DIMENSION A(1),B(1),V(1)
C     GIVEN VECTORS A,B
C     FIND DISTANCE (= UVECN) FROM A TO B.
C     IF DISTANCE NOT ZERO AND K=0, FIND V, A UNIT
C       VECTOR FROM A TOWARD B.
C     OTHERWISE, V=B-A.
C
      D=0.                                                          ,1
      DO 100 I=1,N                                                  ,2
      V(I)=B(I)-A(I)                                                ,3
  100 D=D+V(I)**2
C                                                                   ,4    ,5
      IF (D)500,500,200                                             ,6
  200 D=SQRT(D)                                                     ,7
      IF (K)500,300,500                                             ,8
  300 DO 400 I=1,N                                                  ,9
  400 V(I)=V(I)/D                                                   ,10   ,11
  500 UVECN=D                                                       ,12
      RETURN                                                        ,13
      END                                                           ,14
```

```
      VCROS                                                            07/31/65
  EXTERNAL FORMULA NUMBER   -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      SUBROUTINE VCROS(A,B,C,D,K)
      DIMENSION A(3),B(3),C(3),X(3),Y(3),Z(3)
C
C     FINDS THE CROSS PRODUCT, VECTOR C, OF VECTORS A AND B.
C     C = A X B.  IF K NOT 0 THEN C IS NORMALIZED. =0, NOT.
C     D = MAGNITUDE OF C BEFORE NORMALIZED.
C     ON OUTPUT, K=0 IF SUCCESS, =1 IF NORM. REQUESTED BUT D=0.
C     C MAY BE STORED IN A OR B IF DESIRED.
C
      DO 10 I=1,3                                                      ,1
      X(I)=A(I)                                                        ,2
   10 Y(I)=B(I)
                                                                       ,3     ,4
      Z(1)=X(2)*Y(3)-X(3)*Y(2)                                         ,5
      Z(2)=X(3)*Y(1)-X(1)*Y(3)                                         ,6
      Z(3)=X(1)*Y(2)-X(2)*Y(1)                                         ,7
      E=SQRT (VDOTP(Z,Z))                                              ,8
      D=E                                                              ,9
      IF (K)20,50,20                                                   ,10
   20 IF (E)40,30,40                                                   ,11
   30 K=1                                                              ,12
      GO TO 50                                                         ,13
   40 K=0                                                              ,14
      GO TO 100                                                        ,15
   50 E=1.                                                             ,16
  100 DO 110 I=1,3                                                     ,17
  110 C(I)=Z(I)/E                                                      ,18    ,19
      RETURN                                                           ,20
      END                                                              ,21
```

```
            VDOTN                                                                    07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      FUNCTION VDOTN(A,B,N)
      DIMENSION A(N),B(N)
C     DOT PRODUCT OF TWO VECTORS IN N-SPACE
      VDOTN=0.                                                         ,1
      DO 100 I=1,N                                                     ,2
  100 VDOTN=VDOTN+A(I)*B(I)                                            ,3      ,4
      RETURN                                                           ,5
      END                                                              ,6
```

```
     VDOTP                                                                  07/31/65
 EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      FUNCTION VDOTP(P,Q)
C
      DIMENSION P(3),Q(3)

C     FINDS THE DOT PRODUCT OF VECTORS P AND Q

      VDOTP=P*Q+P(2)*Q(2)+P(3)*Q(3)                                          ,1
      RETURN                                                                 ,2
      END                                                                    ,3
```

```
      SUBROUTINE VEL1(XC,YC,ZC,U,VV,WW)
      ACOSH(U)=2.*ALOG(SQRT(ABS((U+1.)/2.))+SQRT(ABS((U-1.)/2.)))          2     3     4
      COMMON /FLOV1/KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS,NROW,XMACH,SYM
      COMMON /FLOV2/A,B,C,NPART,ALPHAS,THETA,XB ,R,WT,T,TC,SST,CHORD
      COMMON /FLOV3/ T11, TC11
      DIMENSION A(210,3,4),B(210,3,4),C(210,3,4),NROW(2),NPART(210)
      DIMENSION ALPHAS(210),THETA(210)
      DIMENSICN  XB(50),R(50),WT(120)
      DIMENSION T(50),TC(50),SST(210) ,CHORD(210)
      DIMENSION XC(1),YC(1),ZC(1),U(1),AN1(1),SY(2),VV(1),WW(1)
      COMMON /THICK/THKW,ARA
      BT2 = XMACH*XMACH - 1.
      BETA = SQRT(BT2)                                                     7
      PI=3.1415926
      N1 = NBODYS-1
      I =1
      THETAA=ATN1(YC(I),ZC(I))                                             12
      COSTHA=COS(THETAA)                                                   13
      SINTHA=SIN(THETAA)                                                   14
      R2=YC(I)**2+ZC(I)**2
      R1=SQRT(R2)                                                          19
      BR2=BT2*R2
      II=1
      U(I) = 0.
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VT=0.
      VTF = 0.
      VF = 0.
      DUM = 0.
      US1 = 0.
      UD1 = 0.
      VS1 = 0.
      VD1 = 0.
      VTD1 = 0.
      USS = 0.
      UDS = 0.
      VSS = 0.
      VDS = 0.
      VTS = 0.
      TRM2 = SQRT(BR2)                                                     22
      DO 780 J=1,N1
      TRM1 = XC(I)-XB(J)+BETA*R(J)
      TRM = XC(I)-(XB(NBODYS)-BETA*R(NBODYS))
      XI = XB(NBODYS)-(XB(J)-BETA*R(J))-BETA*R(NBODYS)
      IF (R2.EQ.0.) GO TO 770
      TRM3 = TRM2/TRM1
      TRM32=TRM3*TRM3
      IF (TRM3.GT.1..OR.TRM3.LE.0.) GO TO 775
C     SINGULARITY IS INSIDE FOWARD FACING MACH CONE
C     VELOCITY DUE TO QUADRATICALLY VARYING SINGULARITIES
```

```
      TRM4=SQRT(1.-TRM32)                                              42
      TRM5=ACOSH(TRM1/TRM2)
      US=-2.*TRM1*(TRM5-TRM4)*T(J)
      VS= BETA*TRM1*(TRM4/TRM3-TRM3*TRM5)*T(J)
      UD= TRM1**2/R1*(TRM4-TRM32*TRM5)*TC(J)
      VD= -TRM1**3/R1**2*((1.-4.*TRM32)/3.*TRM4+TRM32*TRM5)*TC(J)
      VT=-TRM1**3/R2*((1.+2.*TRM32)/3.*TRM4-TRM32*TRM5)*TC(J)
C     VELOCITY DUE TO TANGENT CONE
      IF (J.NE.1) GO TO 710
      US5 = US
      VS5 = VS
      US = US-ACOSH(TRM1/TRM2)*T11
      VS = VS+BETA*SQRT(1.-TRM32)/TRM3*T11                             51
      USA = US - US5
      VSA = VS - VS5
      UD = UD+BETA*SQRT(1.-TRM32)/TRM3*TC11                            52
      VD = VD-.5*BT2*(SQRT(1.-TRM32)/TRM32+TRM5)*TC11                  53
      VT = VT-.5*BT2*(SQRT(1.-TRM32)/TRM32-ACOSH(TRM1/TRM2))*TC11      54
  710 CONTINUE
  750 CONTINUE
C     SUMMATION OF VELOCITY COMPONENTS
      U(I) = U(I)+US+COSTHA*UD
      VF   = VF   +VS+COSTHA*VD
      VTF    = VTF    + VT*SINTHA
      USAVE = U(I)-USA
      VSAVE = VF-VSA
      WSAVE = VTF

C     FIELD POINT IS INSIDE MACH CONE FROM END OF BODY

      IF (TRM.LE.TRM2) GO TO 775
      TRM6 = SQRT(TRM*TRM-BR2)                                         63
      IF (J.NE.1) GO TO 765
      US1 = T11*(ACOSH(TRM/TRM2)+XI/TRM6)
      UD1 = -COSTHA*TC11*((TRM1*TRM-BR2)/TRM6)/R1
      VS1 = -T11/R1*(TRM6+XI*TRM/TRM6)
      VD1 = DUM
     $+COSTHA*TC11/R2*(BR2*XI/TRM6+BR2/2.*ACOSH(TRM/TRM2)+(TRM1+XI)/2.
     $*TRM6)
      VTD1 =   -SINTHA*TC11/R2*(BR2*.5*ACOSH(TRM/TRM2)-(TRM1+XI)/2.*
     $TRM6)
  765 CONTINUE
      USS = USS+T(J)*(2.*TRM1*ACOSH(TRM/TRM2)-2.*TRM6+XI*XI/TRM6)
      UDS = UDS+COSTHA*TC(J)/R1*(BR2*ACOSH(TRM/TRM2)-(2.*TRM1+XI)/3.*
     $TRM6+(XI*(XI*XI+2.*BR2)-TRM1*(TRM1*TRM1-BR2))/(3.*TRM6))
      VSS = VSS+T(J)/R1*(BR2*ACOSH(TRM/TRM2)-(TRM1+XI)*TRM6-XI*XI
     $*TRM/TRM6)
      VDS = VDS+COSTHA*TC(J)/R2*((TRM1*TRM1+TRM1*XI+XI*XI-2.*BR2)/3.
     $*TRM6+TRM1*BR2*ACOSH(TRM/TRM2)+BR2*(4.*TRM1*XI-2.*TRM1*TRM1
     $+XI*XI+2.*BR2)/(3.*TRM6))
      VTS = VTS-SINTHA*(TRM1*BR2*ACOSH(TRM/TRM2)-(TRM1*TRM1+TRM1*XI
     $+XI*XI+2.*BR2)*TRM6/3.                    )*TC(J)/R2
```

```
      GO TO 775
  770 CONTINUE
  775 CONTINUE
  780 CONTINUE
      U (I) = U(I)+USS+UDS+US1+UD1
      VF = VF + VSS+VDS+VS1+VD1
      VTF = VTF+VTS +VTD1
      SYN=2*(JM-J)-1
      VV(I)=VF    *SINTHA+VTF    *COSTHA
      WW(I)=VF    *COSTHA-VTF    *SINTHA
  790 CONTINUE
  800 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WBX(B,NB,W,NW,EPS,KODE,S,BW,NU)
C
      DIMENSION B(1),W(1),S(2,1),BW(3,1),NU(3)
     1 Q(3,2),R(10),DUM(3)
     2,WGPTS(3,2),UVEC(3),WP(3,2)
     3,PTCNWG(3)
      LOGICAL INTD
C
C     USED BY WBXUL TO FIND THE INTERSECTIONS OF THE UPPER OR LOWER
C       WING PERCENT LINES WITH THE BODY.
C     B IS THE ARRAY OF NB BODY PERCENT LINES.
C     W IS THE ARRAY OF NW WING PERCENT LINES.
C
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, .........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...    ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...    ELEMENTS 3,6,9,...    ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     EPS IS THE TOLERANCE IN LOCATING INTERSECTION POINTS.
C       A WARNING MESSAGE IS WRITTEN IF THE ERROR IS GRFATER THAN EPS.
C     KODE =1 IF THE BODY IS POLYGONAL IN CROSS SECTON, = 2 IF SMOOTH.
C     S IS A SCRATCH ARRAY AT LEAST 2*NB CELLS LONG.  USED FOR 2-DIMENSIONAL
C       CROSS SECTIONS THRU THE BODY.
C     BW IS AN OUTPUT ARRAY OF BODY-WING INTERSECTION POINTS.  IF THE JTH
C       WING PERCENT LINE DOES NOT INTERSECT THE BODY, THEN
C       BW(1,J) = BW(2,J) = BW(3,J) = 1.E+30.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C     INPUTS ARE B,NB,W,NW,EPS,KODE,NU(2),NU(3).  OUTPUTS BW,S,BW,NU(1),NU(3)
C
      DATA SR,DUM/3HWBX,3*1.E+30/
C
      R(10)=1.E-7
C
C     CUT THE BODY WITH A STATION PLANE THRU EACH WING PERCENT POINT IN TURN
C     UNTIL THE INTERSECTICN POINT IS ISOLATED BETWEEN 2 ADJACENT WING POINTS.
C     THEN ITFRATE UNTIL THE X-VALUE OF THE INTERSECTION IS FOUND (WITHIN EPS).
C                                                                               ,1
  100 DO 6000 IW=1,NW
C     FOR EACH WING PERCENT LINE IN W.
```

```
C                                                                          ,2
       INTD=.FALSE.                                                        ,3
       L=IW*3
C      NWP IS NO. OF PTS. IN THE IW PERCENT LINE AND LWP IS THE LOCATION OF THE
C      FIRST POINT IN W.  LWP IS UPDATED IN THE LOOP BELOW.                ,4
       NWP=W(L-1)                                                          ,5
       LWP=W(L)
C                                                                          ,6
 200   DO 1200 K=1,NWP
C      FOR EACH POINT W(LWP) IN THE PERCENT LINE.
C
C      FIND THE HORIZONTAL DISTANCE D FROM POINT W(LWP) TO THE BODY AND THE
C      CORRESPONDING  POINT Q ON THE BODY.                                 ,7
       EPC=AMAX1(ABS(W(LWP)),1.)*1.E-6                                     ,8
       CALL WBXC(B,NB,EPC,W(LWP),KODE,S,D,Q,NU)                            ,9
       IF(NU)250,500,300                                                   ,10
 250   MU=6
C      NU = -5 MEANS BODY IS NOT DEFINED AT STATION X                      ,11
       IF(NU+5)9000,400,9000                                               ,12
 300   MU=7
C      NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY             ,13
       IF(NU-2)310,310,9000                                                ,14
 310   MU=8                                                                ,15
       IF(D)510,9000,510
C      SEE IF PREVIOUS POINT WAS OK.                                       ,16
 400   IF(INTD) GO TO 1250                                                 ,17   ,18   ,19
       GO TO 1000                                                          ,20
 500   IF(D .EQ. 0.) GO TO 1500                                            ,21   ,22   ,23
 510   IF(INTD) GO TO 600                                                  ,24   ,25   ,26
       INTD=.TRUE.                                                         ,27
       GO TO 700
C
C      SEE IF THIS POINT AND PREVIOUS POINT ARE ON OPPOSITE SIDES OF BODY SURFACE   ,28
 600   IF(D*DP)650,700,700
C      OPPOSITE SIDES
C      THE INTERSECTION PT. IS BETWEEN W(LWPP) AND W(LWP)                  ,29
 650   R(4)=D                                                              ,30
       GO TO 1600
C
C      SAME SIDE OR ELSE FIRST TIME WBXC SUCCEEDED.                        ,31
 700   DP=D
C                                                                          ,32
 1000  LWPP=LWP                                                            ,33
       LWP=LWP+3
C                                                                          ,34
 1200  CONTINUE
C                                                                          ,35   ,36
       GO TO 1300
C
C      THE X-VALUE (=W(LWP)) OF THIS WING PERCENT POINT IS OUTSIDE
C      THE RANGE OF BODY X-VALUES.  HOWEVER, THE PREVIOUS WING
C      PERCENT POINT WAS OK.  ONLY 1 OF THE BODY END POINTS CAN
C      LIE BETWEEN W(LWPP) AND W(LWP).  FIND THIS POINT, INTERSECT
C      THE WING PERCENT LINE SEGMENT WITH THE CORRESPONDING
```

```
            WBX                                                                    08/19/65
         EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

C      BODY STATION PLANE TO GET A POINT ON THE WING P.C. LINE.
C      CALL THIS POINT  PTONWG.
C      THE DISTANCE FROM THIS PT. TO THE BODY SURFACE WILL BE IN
C      R(2)=R(4).  THE X-VALUE WILL BE IN R(1)=R(3).
C                                                                          ,37
 1250  CALL WBXD(R,B,NB,KODE,PTONWG,Q,S,W(LWPP),NU)                        ,38
       MU=5                                                                ,39
       IF(NU)9000,1260,9000                                                ,40
 1260  D=R(2)                                                              ,41
       IF(D*DP)1600,1600,1300
C
C      NO INTERSECTION.  LEAVE PERCENT LINE UNCHANGED.  STORE X,Y,Z = 1.E+30
C      AS INTERSECTION POINT.                                              ,42
 1300  KOUT=0                                                              ,43
       CALL TRAV(BW(1,IW),DUM,3)                                           ,44
       GO TO 5000
C
C      WING POINT W(LWP) LIES ON THE BODY SURFACE.                         ,45
 1500  KOUT=K                                                              ,46
       GO TO 4500
C                                                                          ,47
 1600  CONTINUE                                                            ,48
       CALL TRAV(WGPTS,W(LWPP),6)                                          ,49
       IF(MU .NE. 5) GO TO 1650                                            ,50    ,51    ,52
       CALL TRAV(WGPTS(1,2),PTONWG,3)
C      WGPTS CONTAINS THE COORD. OF 2 CONSECUTIVE POINTS ON THE
C      WING PERCENT LINE WHICH LIE ON OPPOSITE SIDES OF THE BODY SURFACE.
C      FIND THE INTERSECTION OF THE LINE SEGMENT WGPTS WITH THE BODY
C      SURFACE AS FOLLOWS,  CHOOSE A POINT (IN WP) WHICH IS SOME
C      DISTANCE ALONG WGPTS FROM ITS 1ST END POINT.  (A UNIT VECTOR
C      ALONG WGPTS WILL BE UVEC).  USE WBXC TO FIND THE HORIZONTAL
C      DISTANCE (IN Y-DIRECTION) FROM WP TO THE BODY SURFACE.
C      ITERATE ON DISTANCE ALONG WGPTS UNTIL WP LIES ON BODY SURFACE.
C
C      USE FRODTA TO TAKE CARE OF THE BOOK-KEEPING FOR THE ITERATION
C      AND TO PREDICT DISTANCES ALONG WGPTS TO USE FOR NEXT ITERATION.
C      FRODTA WILL SET JR = 0 WHEN INTERSECTION IS FOUND.
C                                                                          ,53
 1650  CONTINUE                                                            ,54
       R(1)=0.                                                             ,55
       R(2)=DP                                                             ,56
       R(3)=UVECN(WGPTS,WGPTS(1,2),UVEC,0,3)
C      R(3) = LENGTH OF THE LINE SEGMENT WGPTS
C      R(4) WAS SET TO D PREVIOUSLY                                        ,57
       R(9)=0.                                                             ,58
       MU=9                                                                ,59
       IF(R(3))9000,9000,1700                                              ,60
 1700  MU=10
C                                                                          ,61
       DO 3000 KR=1,100                                                    ,62
       CALL FRODTA(R,JR)
C      R(1) AND R(3) ARE NEW TRIAL VALUES OF DISTANCE ALONG WGPTS
                                                                           ,63
       DO 1900 I=1,2                                                       ,64
```

```
      II=I+I                                                       ,65
      DO 1800 J=1,3                                                ,66
 1800 WP(J,I)=WGPTS(J,I)+R(II-1)*UVEC(J)                           ,67   ,68
      CALL WBXC(B,NB,EPC,WP(1,I),KODE,S,R(I1),Q(1,I),NU)           ,69
C     R(2) AND R(4) ARE CORRESPONDING Y-DISTANCES OF WP FROM BODY SURFACE
      IF(NU .LT. 0  .OR.  NU .GT. 2) GC TO 9000                    ,70   ,71   ,72
 1900 CONTINUE                                                     ,73   ,74
C
      IF(JR)2000,3100,3000                                         ,75
 2000 MU=11                                                        ,76
      GO TO 9000                                                   ,77
C
 3000 CONTINUE
C
C     DID NOT CONVERGE                                             ,78   ,79
      MU=12                                                        ,80
      GO TO 9000                                                   ,81
C
 3100 IF(ABS(R(2))-EPS) 4000,4000,3110                             ,82
 3110 LTAPE=NU(2)                                                  ,83
      NU(3)=NU(3)-1                                                ,84
      IF(LTAPE .LE. 0  .OR.  NU(3) .LE. 0)GO TO 4000               ,85   ,86   ,87
      WRITE(LTAPE,3120)IW,EPS,(WP(I,1),I=1,3)                      ,88   ,89   ,90   ,91   ,92   ,93

 3120 FORMAT(/42H0*** INTERSECTION OF WING PERCENT LINE NO.
     1 I3,40H WITH BODY NOT WITHIN GIVEN TOLERANCE OF
     2 F10.6/27H POINT ON WING HAS COORD. ( 3F10.4,1H) )
C
C     THE INTERSECTION (IN Q) HAS BEEN FOUND AND LIES BETWEEN
C     POINTS K-1 AND K.
 4000 KOUT=K-1
C     THE FIRST KOUT POINTS ARE INSIDE THE BODY AND ARE TO
C     BE REMOVED.
C     STORE THE INTERSECTION POINT.
 4500 CALL TRAV(BW(1,IW),Q,3)                                      ,94
C
C     ADJUST THE W ARRAY TO REFLECT ONLY THAT PART OF WING
C     OUTSIDE BODY.
 5000 CALL WBXA(W,IW,KOUT,BW(1,IW),NU)                             ,95
      MU=4                                                         ,96
      IF (NU)9000,6000,9000                                        ,97
C
 6000 CONTINUE                                                     ,98
      MU=0                                                         ,99   ,100
      GO TO 9900                                                   ,101
C
C     WRITE ERROR MESSAGE.
 9000 LT=NU(2)                                                     ,102
      NU(3)=NU(3)-1                                                ,103
      IF(LT .LE. 0 .OR. NU(3) .LE.0) GO TO 9900                    ,104
      WRITE(LT,9100)MU,NU(1),SR,IW,K                               ,105  ,106  ,107
 9100 FORMAT(6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6/         ,108  ,109  ,110
     , 18H WING PERCENT LINE I5,7H, POINT I5)
 9900 NU=MU                                                        ,111
```

```
      WBX                                                    08/19/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

RETURN                                                                  ,112
END                                                                     ,113
```

```
      SUBROUTINE WBXA(B,I,K,P,NU)
C
      DIMENSION B(1),P(3)
C
C     USED BY WBX TO ADJUST THE B ARRAY OF WING PERCENT LINES
C     TO REMOVE THAT PART OF THE I-TH PERCENT LINE INSIDE BODY.
C     INPUTS ARE B,I,K,P.  OUTPUTS ARE B,NU.
C
      L=3*I
C     KB IS NO. OF POINTS IN I-TH PERCENT LINE.                          ,1
      KB=B(L-1)
C     KC IS STARTING LOCATION OF I-TH PERCENT LINE.                      ,2
      KC=B(L)                                                            ,3
      KO=KC+3*K-1                                                        ,4
      IF (K .GE. KB .OR. K .LT. 0) GO TO 5000                            ,5    ,6    ,7
      IF (I .EQ. 1) GO TO 1000
C     JB IS NO. OF PTS. IN (I-1)TH PERCENT LINE                          ,8    ,9    ,10
      JB=B(L-4)
C     JC IS STARTING LOCATION.                                           ,11
      JC=B(L-3)
C     KN = STARTING LOCATION OF NEW PERCENT LINE                         ,12
      KN=JC+3*JB                                                         ,13
      B(L)=KN                                                            ,14
      GO TO 2000
C                                                                        ,15
 1000 KN=KC                                                              ,16
 2000 NN=KB-K                                                            ,17
      IF (K)2100,2100,2200                                               ,18
 2100 NN=NN-1
C     NN+1 = NO. OF PTS. IN NEW PERCENT LINE.                            ,19
 2200 B(L-1)=NN+1                                                        ,20
      KN=KN-1
C                                                                        ,21
      IF (K)3000,4000,3000
C     INSERT P AS FIRST PT. IN NEW PERCENT LINE.                         ,22
 3000 DO 3100 J=1,3                                                      ,23
      KN=KN+1                                                            ,24
 3100 B(KN)=P(J)
C
C     MOVE LAST PART OF OLD PERCENT LINE UP TO NEW POSITION.             ,25   ,26
 4000 NN=3*NN                                                            ,27
      DO 4100 J=1,NN                                                     ,28
      KN=KN+1                                                            ,29
      KO=KO+1                                                            ,30
 4100 B(KN)=B(KO)
C                                                                        ,31   ,32
      NU=0                                                               ,33
      GO TO 9000
C                                                                        ,34
 5000 NU=1                                                               ,35
 9000 RETURN                                                             ,36
      END                                                                ,37
```

```
      SUBROUTINE WBXC(B,NB,EP,P,KODE,S,D,Q,NU)
C
      DIMENSION B(1),P(3),S(1),Q(3),MU(3)
      DATA MU(2)/0/
C
C     USED BY WBX AND WBXD.  GIVEN A POINT P, CUT THE BODY WITH THE
C     PLANE X = P(1) AND GET BODY SECTION S (POINTS Y,Z).
C     FIND THE HORIZONTAL DISTANCE D FROM P TO S AND THE
C     CORRESP. POINT Q ON S (Q(1)=P(1) AND Q(3)=P(3)).
C     INPUTS-  B,NB,EP,P,KODE.   OUTPUTS-  S,D,Q,NU
C
      X=P(1)                                                     ,1
      Z=P(3)                                                     ,2
C     CUT BODY WITH STATION PLANE
      CALL BCUTX(B,NB,X,EP,NS,S)                                 ,3
      NU=-5                                                      ,4
      IF (NS .LT. 2) GO TO 9000                                  ,5     ,6     ,7
C     INTERPOLATE WITH Z AS INDEPENDENT VARIABLE TO GET
C     CORRESPONDING Y.
      CALL BITURP(S(2),S,2,NS,KODE,Z,Y,MU)                       ,8
      Q(1)=X                                                     ,9
      NU=MU                                                      ,10
      IF(MU)9000,5000,4000                                       ,11
 4000 IF(MU-2)6000,7000,9000                                     ,12
 5000 Q(2)=Y                                                     ,13
      Q(3)=Z                                                     ,14
      GO TO 8000                                                 ,15
 6000 Q(2)=S(2)                                                  ,16
      Q(3)=S(1)                                                  ,17
      GO TO 8000                                                 ,18
 7000 L=NS+NS                                                    ,19
      Q(2)=S(L)                                                  ,20
      Q(3)=S(L-1)                                                ,21

 8000 D=P(2)-Q(2)                                                ,22
C
 9000 RETURN                                                     ,23
      END                                                        ,24
```

```
      SUBROUTINE WBXD(R,B,NB,KODE,P,Q,S,W,NU)
      DIMENSION R(4),B(1),P(3),Q(3),S(2,1),W(3,2),NU(3),C(4),LP(2)
C
C     USED BY WBX IF NECESSARY TO PROCESS A WING PERCENT POINT
C     WITH X-VALUE OUTSIDE THE RANGE OF BODY PERCENT LINE X-VALUES
C
C     W(I,1) IS A WING PERCENT POINT WHICH WAS PROCESSED OK BY WBX.
C     W(I,2) IS NEXT POINT, WHICH IS OUTSIDE BODY RANGE.
C     OTHER ARGUMENTS EXPLAINED IN WBX
C     RECALL THAT B IS HEADED BY TABLE OF CONTENTS.
C     S IS SCRATCH ARRAY FOR WBXC.
C
C     INPUTS  ARE B,NB,KODE,W,NU(2),NU(3)
C     OUTPUTS ARE P,Q,S,NU(1),NU(3)
C
      DATA SR,C/4HWBXD,-1.,3*0./
C                                                                  ,1
      NU=0                                                         ,2
      IF=B(3)                                                      ,3
      ISEC=B(6)                                                    ,4
      BXF=B(IF)
      BXL=B(ISEC-3)                                                ,5
C     BXF AND BXL ARE EXTREME X-VALUES OF BODY PERCENT LINE 1       ,6
      WXF=W(1,1)
      WXL=W(1,2)                                                   ,7
C     TEST TO SEE IF BXF IN RANGE OF WING POINTS
      IF((BXF-WXF)*(BXF-WXL))100,100,200                           ,8
C     BXF WITHIN RANGE                                             ,9
  100 R(1)=BXF
      GO TO 300                                                    ,10
C     BXF NOT IN RANGE.  ASSUME BXL IS.
  200 R(1)=BXL                                                     ,11
C                                                                  ,12
  300 R(3)=R(1)
      C(4)=R(1)                                                    ,13
C     C IS A STATION PLANE
      CALL POLXN(C,W,3,2,0.,1,P,LP,L)                              ,14
C     P IS INTERSECTION OF PLANE AND LINE SEGMENT W.               ,15
      MU=1
      IF (L .NE. 1) GO TO 8000
C
C     FIND HORIZONTAL DISTANCE FROM P TO BODY AND
C     CORRESPONDING POINT Q ON BODY.                               ,16    ,17    ,18
      CALL WBXC(B,NB,0.,P,KODE,S,R(2),Q,NU)                        ,19
      MU=2                                                         ,20
      IF(NU)8000,1000,500
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY       ,21
  500 IF(NU-2)600,600,8000                                         ,22
  600 IF(R(2)) 1000,8000,1000
C                                                                  ,23
 1000 CONTINUE                                                     ,24
      MU=0                                                         ,25
      R(4)=R(2)                                                    ,26
```

```
      GO TO 9000
C
C     WRITE ERROR MESSAGE.                                           ,27
 8000 LT=NU(2)                                                       ,28
      NU(3)=NU(3)-1                                                  ,29
      IF(LT .LE. 0  .OR.  NU(3) .LE. 0) GO TO 9000                   ,30   ,31   ,32
      WRITE(LT,8100)MU,NU(1),SR                                      ,33   ,34   ,35
 8100 FORMAT(6H0ERROR I6,6H, CODE I6,15H IN SUBROUTINE A6)
C
 9000 CONTINUE                                                       ,36
      NU=MU                                                          ,37
      RETURN                                                         ,38
      END                                                            ,39
```

```
      SUBROUTINE WBXUL(B,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU)
C
      DIMENSION B(1),DAT(2),LOK(3,9),NU(3),UPLO(2),PT(3),
     ,BTITLE(12),WTITLE(12),BAXIS(2)
C
      LOGICAL LGDEF(3,6)
      DATA SR,UPLO(1)/5HWBXUL,12H UPPER LOWER/,DEGRAD/0206712273406/
     ,,EP/1.E-5/
C
C       USED BY GEOMD TO FIND THE INTERSECTIONS OF THE UPPER AND/OR
C       LOWER WING PERCENT LINES WITH THE BODY SURFACE, AND TO
C       TRUNCATE THE WING PERCENT LINES SO THAT THEY REPRESENT ONLY
C       THAT PART OF THE WING OUTSIDE THE BODY.
C
C       INPUTS ARE B,LI,LO,DAT,LGDEF,LOK,BTITLE,WTITLE,BAXIS,NU
C       OUTPUTS ARE B,LGDEF,LOK,NU
C
C       RECORD THAT THE WING-BODY INTERSECTION HAS BEEN REQUESTED
 100  LGDEF(1,5)=LGDEF(1,1)                                         ,1
      LGDEF(1,6)=LGDEF(1,2)
C                                                                   ,2
      NU=0
C      SEE IF BODY AND AT LEAST 1 SURFACE OF WING ARE DEFINED.      ,3
      IF ((LGDEF(2,1) .OR. LGDEF(2,2)) .AND. LGDEF(2,3)) GO TO 300  ,4    ,5    ,6
      WRITE (LO,210)                                                ,7    ,8
 210  FORMAT(47H1WING-BODY INTERSECTION CANNOT BE FOUND BECAUSE
     1/44H EITHER THE BODY OR THE WING IS NOT DEFINED. )
      NU=1                                                          ,9
      GO TO 9000
C                                                                   ,10
 300  READ(LI,310)WBKODE,EPS                                        ,11   ,12   ,13
 310  FORMAT(6F10.0)
      IF(ABS(EPS)-EP) 320,330,330                                   ,14
 320  EPS=EP                                                        ,15
 330  CONTINUE                                                      ,16
      KODE=WBKODE
C      KODE = 1 IF BODY IS POLYGONAL IN SHAPE, =2 IF SMOOTH.
                                                                    ,17
      WRITE(LO, 340)KODE,EPS                                        ,18   ,19   ,20
 340  FORMAT(/71H0FIND THE INTERSECTION OF EACH WING PERCENT LINE WITH T
     1HE BODY SURFACE. // 1H0 10X 6HCODE = I3/1H0 10X11HTOLERANCE =F7.4/
     2/48H0LET P BE A POINT WHICH LIES ON THE BODY SURFACE /
     356H AND Q A POINT ON A LINE SEGMENT CONNECTING TWO ADJACENT /
     472H WING PERCENT LINE POINTS, SUCH THAT P AND Q HAVE THE SAME X-CO
     5ORDINATE. /
     649H0THE PROGRAM ITERATES UNTIL P AND Q HAVE THE SAME /
     742H (OR NEARLY THE SAME) Y AND Z COORDINATES. /
     834H THEN P IS THE INTERSECTION POINT. /
     952H0IF P AND Q DIFFER BY MORE THAN THE GIVEN TOLERANCE, /
     A30H A WARNING MESSAGE IS WRITTEN. /
     B69H0THE SHAPE OF THE BODY SURFACE BETWEEN MERIDIAN LINES (IN ANY S
     CECTION /
     D59H PARALLEL TO THE YZ PLANE) IS DETERMINED BY THE GIVEN CODE. /
```

```
          WBXUL                                                                07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      E5X65HIF CODE = 1, LINEAR INTERPOLATION IS USED BETWEEN MERIDIAN LI
      FNES.
      G5X47HIF CODE = 2, BIQUADRATIC INTERPOLATION IS USED.
C
      NB=LOK(2,5)
C      NB IS NO. OF BODY PERCENT LINES.  RESERVE STORAGE FOR SCRATCH
C      ARRAY OF LENGTH 2*NB FOR SECTIONS THRU BODY.  STARTS IN B(NAMS).     ,21
      MU=2                                                                  ,22
      IF (IRSERV(NB+NB,B,NAMS))8000,8000,400
C      PACK BUFFER                                                          ,23
 400  IF (IPACK(B))8000,500,500                                             ,24
 500  LB=LOK(1,5)
C      BODY PERCENT LINES START IN B(LB)
C
C      MAIN LOOP ON K IS FOR UPPER WING, THEN LOWER WING.
C                                                                           ,25
      DO 6000 K=1,2
C
C      SEE IF THIS WING SURFACE IS DEFINED                                  ,26
      IF (.NOT. LGDEF(2,K)) GO TO 6000
C      WING DEFINED.  FIND NW = NO. OF WING PERCENT LINES.                  ,27   ,28   ,29
      NW=LOK(2,K)
C
C      RESERVE STORAGE FOR WING-BODY INTERSECTION POINTS                    ,30
      MU=3                                                                  ,31
      NWBE=3*NW                                                             ,32
      K7=K+7                                                                ,33
      IF (IRSERV(NWBE,B,LOK(1,K7)))8000,8000,700                            ,34
 700  LOK(3,K7)=NWBE                                                        ,35
      LWB=LOK(1,K7)                                                         ,36
      LB=LOK(1,5)                                                           ,37
      NAMS=NAMS
C      FIND LW = LOCATION OF WING PERCENT LINES (START IN B(LW))            ,38
      LW=LOK(1,K)
C
C      FIND INTERSECTIONS AND TRUNCATE WING PERCENT LINES                   ,39
      CALL WBX(B(LB),NB,B(LW),NW,EPS,KODE,B(NAMS),B(LWB),NU)                ,40
      MU=4                                                                  ,41
      IF (NU)8000,800,8000
C
C      WRITE HEADING                                                        ,42
 800  WRITE (LO,810)BTITLE,WTITLE,DAT,UPLO(K),BAXIS                         ,43   ,44   ,45
 810  FORMAT(1H1 12A6/1H 12A6/54X 2A6/18H INTERSECTIONS OF A6,
     ,43H WING PERCENT CHORD LINES WITH BODY SURFACE /
     ,41H RHO-THETA ORIGIN IS LOCATED AT (Y,Z) = ( F9.4,1H,F9.4,1H)/
     ,13HONO.  PERCENT 6X 1HX 9X 1HY 9X 1HZ 10X 3HRHO 6X 5HTHETA//)
C
C      WRITE THE INTERSECTION POINTS
      IPCT=LW                                                               ,46
      IWBX=LWB
C                                                                           ,47
      DO 1000 J=1,NW                                                        ,48
      PCT=B(IPCT)                                                           ,49
      IPCT=IPCT+3                                                           ,50
```

```
           WBXUL                                                                  07/31/65
        EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      CALL TRAV(PT,B(IWBX),3)                                                    ,51
      IWBX=IWBX+3                                                                ,52
      IF (PT .GT. 1.E+29) GO TO 900                                              ,53    ,54    ,55
      DY=PT(2)-BAXIS(1)                                                          ,56
      DZ=PT(3)-BAXIS(2)                                                          ,57
      RHO=SQRT(DY*DY+DZ*DZ)                                                      ,58
      IF(RHO)820,820,830                                                         ,59
 820  THETA=0.                                                                   ,60
      GO TO 850                                                                  ,61
 830  THETA=ATN1(ABS(DY),DZ)*DEGRAD                                              ,62
 850  WRITE(LO,860)J,PCT,PT,RHO,THETA                                            ,63    ,64    ,65
 860  FORMAT(1H I2,F10.4,1X 3F10.4,2X 2F10.4)
      GO TO 1000                                                                 ,66
 900  WRITE(LO,910)J,PCT                                                         ,67    ,68    ,69
 910  FORMAT(1H I2,F10.4,6X 15HNO INTERSECTION)
1000  CONTINUE
C
C      RECORD THAT THE INTERSECTIONS HAVE BEEN FOUND                             ,70    ,71
2000  LGDEF(2,K+4)=.TRUE.
C                                                                                ,72
      DO 2100 J=1,2                                                              ,73
2100  LOK(J,K+2)=LOK(J,K)
C                                                                                ,74    ,75
      NLAST=B(IPCT-2)                                                            ,76
      LLAST=B(IPCT-1)                                                            ,77
      LOK(3,K+2)=LLAST+3*NLAST-1
C
C                                                                                ,78
6000  CONTINUE
C                                                                                ,79    ,80
      MU=0                                                                       ,81
      GO TO 8500
C
C      WRITE ERROR MESSAGE                                                       ,82
8000  KE=MERR(NU,NU+1,MU,SR)
C                                                                                ,83
8500  NU=MU
C                                                                                ,84
9000  RETURN                                                                     ,85
      END                                                                        ,86
```

```
      SUBROUTINE WBXX(B,NB,W,NW,EPS,KODE,S,BW,NU)
C
C     SAME AS WBX, EXCEPT DOES NOT STORE TRUNCATED LINES IN B.

      DIMENSION B(1),W(1),S(2,1),BW(3,1),NU(3),
     1 Q(3,2),R(10),DUM(3)
     2,WGPTS(3,2),UVEC(3),WP(3,2)
     3,PTCNWG(3)
      LOGICAL INTO
C
C     USED BY TFLAT TO FIND THE INTERSECTIONS OF WING PERCENT
C     LINES WITH THE BODY SURFACE.

C     B IS THE ARRAY OF NB BODY PERCENT LINES.
C     W IS THE ARRAY OF NW WING PERCENT LINES.
C
C     A PERCENT LINE OF N POINTS HAS 3*N ELEMENTS
C             X1,Y1,Z1, X2,Y2,Z2, .........., XN,YN,ZN
C     THE FIRST 3*J CELLS OF AN ARRAY B OF J PERCENT LINES FORMS A TABLE OF
C     CONTENTS TO THE REST OF THE ARRAY.  ELEMENTS 1,4,7,...   OF B ARE LABELS
C     FOR PERCENT LINES 1,2,3,...    ELEMENTS 2,5,8,... ARE THE NUMBER OF
C     3-DIMENSIONAL POINTS IN PERCENT LINES 1,2,3,...   ELEMENTS 3,6,9,...   ARE
C     THE STARTING LOCATIONS IN B OF PERCENT LINES 1,2,3,...
C     FOR EXAMPLE, IF B(7) = 30., B(8) = 12. AND B(9) = 49., THEN THE PERCENT
C     LINE WITH THE LABEL 30. CONTAINS 12 POINTS AND STARTS IN B(49).
C
C     EPS IS THE TOLERANCE IN LOCATING INTERSECTION POINTS.
C       A WARNING MESSAGE IS WRITTEN IF THE ERROR IS GREATER THAN EPS.
C     KODE =1 IF THE BODY IS POLYGONAL IN CROSS SECTON, = 2 IF SMOOTH.
C     S IS A SCRATCH ARRAY AT LEAST 2*NB CELLS LONG.  USED FOR 2-DIMENSIONAL
C       CROSS SECTIONS THRU THE BODY.
C     BW IS AN OUTPUT ARRAY OF BODY-WING INTERSECTION POINTS.  IF THE JTH
C       WING PERCENT LINE DOES NOT INTERSECT THE BODY, THEN
C       BW(1,J) = BW(2,J) = BW(3,J) = 1.E+30.
C
C     NU IS AN ERROR ARRAY WHICH IS PASSED DOWN TO LOWER LEVEL SUBROUTINES.
C     NU(1), AN OUTPUT, IS ZERO FOR SUCCESS, NON-ZERO FOR FAILURE.
C     NU(2), AN INPUT, MAY BE INTERPRETED AS A  TAPE NUMBER ON WHICH TO
C       WRITE AN ERROR MESSAGE, IF GREATER THAN ZERO.
C     NU(3), BOTH AN INPUT AND AN OUTPUT, IS AN ERROR COUNTER AND ERROR MESSAGE
C       LIMITER.  SOME LOWER LEVEL SUBROUTINES MAY SUBTRACT 1 FROM NU(3) IF AN
C       ERROR OCCURS AND THEN WRITE AN ERROR MESSAGE IF NU(3) AND NU(2) ARE BOTH
C       GREATER THAN ZERO.
C
C     INPUTS ARE B,NB,W,NW,EPS,KODE,NU(2),NU(3). OUTPUTS BW,S,BW,NU(1),NU(3)
C
      DATA SR,DUM/3HWBX,3*1.E+30/
C
      R(10)=1.E-7
C
C     CUT THE BODY WITH A STATION PLANE THRU EACH WING PERCENT POINT IN TURN
C     UNTIL THE INTERSECTION POINT IS ISOLATED BETWEEN 2 ADJACENT WING POINTS.
C     THEN ITERATE UNTIL THE X-VALUE OF THE INTERSECTION IS FOUND (WITHIN EPS).
```

```
C                                                                               ,1
 100  DO 6000 IW=1,NW
C     FOR EACH WING PERCENT LINE IN W.
C                                                                               ,2
      INTD=.FALSE.                                                              ,3
      L=IW*3
C     NWP IS NO. OF PTS. IN THE IW PERCENT LINE AND LWP IS THE LOCATION OF THE
C     FIRST POINT IN W.  LWP IS UPDATED IN THE LOOP BELOW.                      ,4
      NWP=W(L-1)                                                                ,5
      LWP=W(L)
C                                                                               ,6
 200  DO 1200 K=1,NWP
C     FOR EACH POINT W(LWP) IN THE PERCENT LINE.
C
C     FIND THE HORIZONTAL DISTANCE D FROM POINT W(LWP) TO THE BODY AND THE
C     CORRESPONDING  POINT Q ON THE BODY.                                       ,7
      EPC=AMAX1(ABS(W(LWP)),1.)*1.E-6                                           ,8
      CALL WBXC(B,NB,EPC,W(LWP),KODE,S,D,Q,NU)                                  ,9
      IF(NU)250,500,300                                                         ,10
 250  MU=6
C     NU = -5 MEANS BODY IS NOT DEFINED AT STATION X                            ,11
      IF(NU+5)9000,400,9000                                                     ,12
 300  MU=7
C     NU = 1 OR 2 MEANS WING POINT IS ABOVE OR BELOW THE BODY                   ,13
      IF(NU-2)310,310,9000                                                      ,14
 310  MU=8                                                                      ,15
      IF(D)510,9000,510
C     SEE IF PREVIOUS POINT WAS OK.                                             ,16
 400  IF(INTD) GO TO 1250                                                       ,17    ,18    ,19
      GO TO 1000                                                                ,20
 500  IF(D .EQ. 0.) GO TO 1500                                                  ,21    ,22    ,23
 510  IF(INTD) GO TO 600                                                        ,24    ,25    ,26
      INTD=.TRUE.                                                               ,27
      GO TO 700
C
C     SEE IF THIS POINT AND PREVIOUS POINT ARE ON OPPOSITE SIDES OF BODY SURFACE ,28
 600  IF(D*DP)650,700,700
C     OPPOSITE SIDES
C     THE INTERSECTION PT. IS BETWEEN W(LWPP) AND W(LWP)                        ,29
 650  R(4)=D                                                                    ,30
      GO TO 1600
C
C     SAME SIDE OR ELSE FIRST TIME WBXC SUCCEEDED.                              ,31
 700  DP=D
C                                                                               ,32
 1000 LWPP=LWP                                                                  ,33
      LWP=LWP+3
C                                                                               ,34
 1200 CONTINUE
C                                                                               ,35    ,36
      GO TO 1300
C
C     THE X-VALUE (=W(LWP)) OF THIS WING PERCENT POINT IS OUTSIDE
C     THE RANGE OF BODY X-VALUES.  HOWEVER, THE PREVIOUS WING
```

```
C     PERCENT POINT WAS OK.  ONLY 1 OF THE BODY END POINTS CAN
C     LIE BETWEEN W(LWPP) AND W(LWP).  FIND THIS POINT, INTERSECT
C     THE WING PERCENT LINE SEGMENT WITH THE CORRESPONDING
C     BODY STATION PLANE TO GET A POINT ON THE WING P.C. LINE.
C     CALL THIS POINT  PTONWG.
C     THE DISTANCE FROM THIS PT. TO THE BODY SURFACE WILL BE IN
C     R(2)=R(4).  THE X-VALUE WILL BE IN R(1)=R(3).
C                                                                          ,37
 1250 CALL WBXD(R,B,NB,KODE,PTONWG,Q,S,W(LWPP),NU)                         ,38
      MU=5                                                                 ,39
      IF(NU)9000,1260,9000                                                 ,40
 1260 D=R(2)                                                               ,41
      IF(D*DP)1600,1600,1300
C
C     NO INTERSECTION.  LEAVE PERCENT LINE UNCHANGED.  STORE X,Y,Z = 1.E+30
C     AS INTERSECTION POINT.                                               ,42
 1300 KOUT=0                                                               ,43
      CALL TRAV(BW(1,IW),DUM,3)                                            ,44
      GO TO 5000
C
C     WING POINT W(LWP) LIES ON THE BODY SURFACE.                          ,45
 1500 KOUT=K                                                               ,46
      GO TO 4500
C                                                                          ,47
 1600 CONTINUE                                                             ,48
      CALL TRAV(WGPTS,W(LWPP),6)                                           ,49
      IF(MU .NE. 5) GO TO 1650                                             ,50   ,51   ,52
      CALL TRAV(WGPTS(1,2),PTONWG,3)
C     WGPTS CONTAINS THE COORD. OF 2 CONSECUTIVE POINTS ON THE
C     WING PERCENT LINE WHICH LIE ON OPPOSITE SIDES OF THE BODY SURFACE.
C     FIND THE INTERSECTION OF THE LINE SEGMENT WGPTS WITH THE BODY
C     SURFACE AS FOLLOWS,  CHOOSE A POINT (IN WP) WHICH IS SOME
C     DISTANCE ALONG WGPTS FROM ITS 1ST END POINT.  (A UNIT VECTOR
C     ALONG WGPTS WILL BE UVEC).  USE WBXC TO FIND THE HORIZONTAL
C     DISTANCE (IN Y-DIRECTION) FROM WP TO THE BODY SURFACE.
C     ITERATE ON DISTANCE ALONG WGPTS UNTIL WP LIES ON BODY SURFACE.
C
C     USE FROOTA TO TAKE CARE OF THE BOOK-KEEPING FOR THE ITERATION
C     AND TO PREDICT DISTANCES ALONG WGPTS TO USE FOR NEXT ITERATION.
C     FROOTA WILL SET JR = 0 WHEN INTERSECTION IS FOUND.
C                                                                          ,53
 1650 CONTINUE                                                             ,54
      R(1)=0.                                                              ,55
      R(2)=DP                                                              ,56
      R(3)=UVECN(WGPTS,WGPTS(1,2),UVEC,0,3)
C     R(3) = LENGTH OF THE LINE SEGMENT WGPTS
C     R(4) WAS SET TO D PREVIOUSLY                                         ,57
      R(9)=0.                                                              ,58
      MU=9                                                                 ,59
      IF(R(3))9000,9000,1700                                               ,60
 1700 MU=10
C                                                                          ,61
      DO 3000 KR=1,100
      CALL FROOTA(R,JR)
```

```
       WBXX                                                              08/19/65
     EXTERNAL FORMULA NUMBER   -    SOURCE STATEMENT    -   INTERNAL FORMULA NUMBER(S)

C      R(1) AND R(3) ARE NEW TRIAL VALUES OF DISTANCE ALONG WGPTS
                                                                           ,63
       DO 1900 I=1,2                                                       ,64
       II=I+I                                                              ,65
       DO 1800 J=1,3                                                       ,66
 1800  WP(J,I)=WGPTS(J,1)+R(II-1)*UVEC(J)                                  ,67   ,68
       CALL WBXC(B,NB,EPC,WP(1,I),KODE,S,R(II),Q(1,I),NU)
C      R(2) AND R(4) ARE CORRESPONDING Y-DISTANCES OF WP FROM BODY SURFACE ,69
       IF(NU .LT. 0  .OR.  NU .GT. 2) GO TO 9000                           ,70   ,71   ,72
 1900  CONTINUE
C                                                                          ,73   ,74
       IF(JR)2000,3100,3000                                                ,75
 2000  MU=11                                                               ,76
       GO TO 9000
C                                                                          ,77
 3000  CONTINUE
C
C      DID NOT CONVERGE                                                    ,78   ,79
       MU=12                                                               ,80
       GO TO 9000
C                                                                          ,81
 3100  IF(ABS(R(2))-EPS) 4000,4000,3110                                    ,82
 3110  LTAPE=NU(2)                                                         ,83
       NU(3)=NU(3)-1                                                       ,84
       IF(LTAPE .LE. 0  .OR.  NU(3) .LE. 0)GO TO 4000                      ,85   ,86   ,87
       WRITE(LTAPE,3120)IW,EPS,(WP(I,1),I=1,3)                             ,88   ,89   ,90   ,91   ,92   ,93

 3120  FORMAT(/42HO*** INTERSECTION OF WING PERCENT LINE NO.
      1 I3,40H WITH BODY NOT WITHIN GIVEN TOLERANCE OF
      2 F10.6/27H POINT ON WING HAS COORD. ( 3F10.4,1H) )
C
C      THE INTERSECTION (IN Q) HAS BEEN FOUND AND LIES BETWEEN
C      POINTS K-1 AND K.
 4000  KOUT=K-1
C      THE FIRST KOUT POINTS ARE INSIDE THE BODY AND ARE TO
C      BE REMOVED.
C      STORE THE INTERSECTION POINT.                                       ,94
 4500  CALL TRAV(BW(1,IW),Q,3)
C                                                                          ,95
 5000  CONTINUE
C                                                                          ,96
 6000  CONTINUE                                                            ,97   ,98
       MU=0                                                                ,99
       GO TO 9900
C
C      WRITE ERROR MESSAGE.                                                ,100
 9000  LT=NU(2)                                                            ,101
       NU(3)=NU(3)-1                                                       ,102
       IF(LT .LE. 0 .OR. NU(3) .LE.0) GO TO 9900                           ,103  ,104  ,105
       WRITE(LT,9100)MU,NU(1),SR,IW,K                                      ,106  ,107  ,108
 9100  FORMAT(6HOERROR I6,6H, CODE I6,15H IN SUBROUTINE A6/
      , 18H WING PERCENT LINE I5,7H, POINT I5)
 9900  NU=MU                                                               ,109
       RETURN                                                              ,110
```

```
EXTERNAL FORMULA NUMBER  -  SOURCE STATEMENT  -  INTERNAL FORMULA NUMBER(S)

END                                                                   ,111
```

```
      SUBROUTINE WING
      COMMON DATE(2),NTAPE1,ND1,NTAPE2,ND2(3),NREAD,
     1NWRITE,NBODY,NWING,XMACH,SYM,KACE
      COMMON /COM1/ KODEB,KODEW,KODEWU,KODEI,KODEC,XPER,YPER,KOPTB,KOPTW
     1,KOPTF,NUMS,KTYPE,KSTART,KEND,KINT,XI(16),YI(16),ZI(16),NPER,NPER1
     2,NPLANE,NPLN1
      COMMON /COM2/IJ,NPTS(15),X(90,16),Y(90,16),Z(90,16),KPNT,VALUE(5),
     1YCEPT(16),SLOPE,KPANEL(2,15),XCOR(16,16),YCOR(16,16),ZCOR(16,16),X
     2INT(2,15),YINT(2,15),ZINT(2,15),XCEN(15,15),YCEN(15,15),ZCEN(15,15
     3),XCON(15,15),YCON(15,15),ZCON(15,15),AREA(15,15),ARAT(15,15),THET
     4A(15,15),ALPHA(15,15),CHORD(15,15),XFOIL(16,25,2),ZFOIL(16,25,2),X
     5NUM(16,2),XTAB1(25),ZTAB1(25),XTAB2(25),ZTAB2(25)
      DIMENSION ND3(9455)
      EQUIVALENCE (ND3(1),NPTS(1))

C     CONTROL PROGRAM FOR WING PANELING SUBROUTINES
      DATA NAME/6HDEFINE/

C     DEFINE CONSTANTS
C     NWING IS NUMBER OF WING PANELS     KODEW.
C     KODEWU ARE INPUT ERROR CODES
C     IF KODEW SET = 1 IN INPUTW THEN TAPE ERROR
C     ENCOUNTERED ON SCRATCH TAPE     IF KODEWU
C     SET = 1, ERROR ENCOUNTERED ON INPUT CARDS
C     FOR THICK WING (OR CAMBERED SURFACE)
 10   NWING=0                                                       ,1
      KODEW=0                                                       ,2
      KODEWU=0                                                      ,3
      IJ=0

C     INITIALIZE COMMON ARRAYS                                      ,4
      DO 20 I=1,9455                                                ,5
 20   ND3(I)=0

C     CALL SUBROUTINES                                              ,6    ,7
 30   CALL INPUTW                                                   ,8
      IF (KODEW) 200,40,110                                         ,9
 40   CALL CRNRW                                                    ,10
      IF (KODEW) 200,45,100                                         ,11
 45   CALL AREAP(XCOR,YCOR,ZCOR,AREA,ARAT,NPLN1,NPER1)              ,12
      CALL CENTRD (XCOR,YCOR,ZCOR,ARAT,XCEN,YCEN,ZCEN,
     1NPLN1,NPER1)                                                  ,13
      CALL CNTRLW                                                   ,14
      CALL CHORDW                                                   ,15
      IF (KODEW) 200,50,100                                         ,16
 50   IJ=IJ+1                                                       ,17
      CALL INPUTW                                                   ,18
      IF (KOPTF) 200,105,60                                         19
 60   IF (KODEWU) 200,80,120                                        ,20
   80 CALL SLOPEW                                                   ,21
      IF (KODEW) 200,105,100                                        ,22
 100  WRITE (NWRITE,920)                                            ,23   ,24
 105  CALL OUTPTW                                                   ,25
```

```
      EXTERNAL FORMULA NUMBER  -  SOURCE STATEMENT  -  INTERNAL FORMULA NUMBER(S)

      IF (KODEW) 150,150,200                                        ,26
110   WRITE (NWRITE,900)                                            ,27   ,28
      GO TO 200                                                      29
120   WRITE (NWRITE,910)                                            ,30   ,31
      GO TO 200                                                     ,32
150   RETURN                                                        ,33
200   READ (NREAD,890) KARD                                          34   ,35   ,36
      IF (KARD-NAME) 200,210,200                                    ,37
210   BACKSPACE NREAD                                               ,38
      KODEW=-1                                                       39
      GO TO 150                                                     ,40
890   FORMAT (A6)
900   FORMAT (1H1,9X,31HERROR MESSAGE - SUBROUTINE WING/10X,
     131HIMPROPER PLANAR WING DEFINITION)
910   FORMAT (1H1,9X,31HERROR MESSAGE - SUBROUTINE WING/10X,
     152HIMPROPER THICK WING (OR CAMBERED SURFACE) DEFINITION)
920   FORMAT (1H1,9X,13HERROR MESSAGE/10X,
     149HPANELLING DATA NOT COMPLETED DUE TO PROGRAM ERROR/10X,
     238HINCOMPLETE PANELLING DATA TO BE OUTPUT)
      END                                                           ,41
```

```
      SUBROUTINE WING1(B,LI,LO,DAT,LLETE,LOGUL,NAF,NPL,LAFN,LPCT,KOD,
     1 CHD,EPS,TITLE,NU)
      DIMENSION B(1),DAT(2),KOD(2),NU(3),NLTEP(4),LPLF(2),HLETE(2,2),
     1 TITLE(12)
      LOGICAL LOGUL(2)
C     WING1 IS USED BY GEOMD.  IT READS DATA CARDS WHICH DEFINE
C     A WING, CHECKS THE DATA, PUTS IT IN STANDARD FORM FOR
C     LATER USE BY WING2 AND GEOMD.
C
C     B IS BUFFER IN WHICH DATA IS STORED
C     LOGUL(1) = TRUE IF UPPER WING IS DEFINED
C     LOGUL(2) = TRUE IF LOWER WING IS DEFINED
C     NAF = NO. OF AIRFOILS
C     NPL = NO. OF PERCENT LINES DESIRED
C     LAFN.   A TABLE IS STORED IN B(LAFN) WHICH HAS THE
C          LOCATIONS IN B WHERE AIRFOILS ARE STORED, AND
C          NO. OF POINTS IN EACH AIRFOIL.
C     LLETE.   THE LEADING AND TRAILING EDGE POINTS OF EACH
C           AIRFOIL ARE STORED IN B(LLETE).
C     LPCT.    THE PERCENTS AT WHICH PERCENT LINE ARE TO BE
C          GENERATED ARE STORED IN B(LPCT).
C     NU IS ERROR INDICATOR.
C
      DATA HLETE(1,1),HLETE(1,2),WING/7HLEADING,8HTRAILING,5HWING1/,
     .HGIVEN,HUSED/6HGIVEN),6HUSED) /,MAX/150/
C
 10   LOGUL(1)=.TRUE.                                          ,1
      LOGUL(2)=.TRUE.                                          ,2
      NU=0
C
C     READ AND WRITE TITLE CARD (CARD 1)                       ,3
      READ (LI,310)TITLE                                       ,4    ,5    ,6
 310  FORMAT(13A6,A2)
      WRITE (LO,320)DAT,TITLE                                  ,7    ,8    ,9
 320  FORMAT(16H1WING DEFINITION 38X,2A6/1H014A6)
C
C     READ CARDS 2,3
      READ (LI,400)PNLE,PNTE,AFN,PLN,WUL,CHD,PCODE,ACODE,EPS   ,10   ,11   ,12
 400  FORMAT(6F10.0)
      NLTEP(1)=PNLE                                            ,13
      NLTEP(2)=PNTE                                            ,14
      NAF=AFN                                                  ,15
      NPL=PLN                                                  ,16
      NLTEP(3)=NAF                                             ,17
      NLTEP(4)=NPL                                             ,18
      WRITE (LO,410)NLTEP                                      ,19   ,20   ,21
 410  FORMAT(  /17H0THE PLANFORM HAS I3,28H POINTS ON THE LEADING EDGE,
     ./I20,29H POINTS ON THE TRAILING EDGE.
     ./I4,20H AIRFOILS ARE GIVEN.
     .  I4,35H PERCENT LINES ARE TO BE GENERATED.)
C
      KOD(1)=ACODE                                             ,22
      KOD(2)=PCODE                                             ,23
```

```
      IF(EPS)411,411,412                                        ,24
411   EPS=.01                                                   ,25
412   WRITE(LO,413)PCODE,ACODE,EPS,CHD                          ,26   ,27   ,28
413   FORMAT( 8HOPCODE = F3.0, 9H, ACODE = F3.0, 7H, EPS = F10.6,
     X 7H, CHD = F7.4)

420   IF (WUL*(WUL-2.))440,450,430
C     WUL NOT 0.,1., OR 2.                                      ,29
430   MU=1                                                      ,30
      GO TO 9000
C     UPPER WING                                                ,31
440   LOGUL(2)=.FALSE.                                          ,32
      WRITE(LO,441)                                             ,33   ,34
441   FORMAT(30H UPPER WING SURFACE IS DEFINED)
      GO TO 480
C     WUL = 0. OR 2.                                            ,35
450   IF (WUL)470,470,460
C     LOWER WING                                                ,36
460   LOGUL(1)=.FALSE.                                          ,37
      WRITE(LO,461)                                             ,38   ,39
461   FORMAT(30H LOWER WING SURFACE IS DEFINED)
      GO TO 480
C     UPPER AND LOWER WING                                      ,40
470   WRITE(LO,471)                                             ,41   ,42
471   FORMAT(42H UPPER AND LOWER WING SURFACES ARE DEFINED)
C
480   CONTINUE
C
C     CHECK FOR REASONABLE VALUES                               ,43
500   DO 520 I=1,4                                              ,44
      I1=NLTEP(I)-1                                             ,45
      IF (I1*(I1-MAX))520,510,510
C     WRITE ERROR MESSAGE  NO. 2 AND GO ON                      ,46
510   IERR=MERR(0,1,2,WING)                                     ,47
520   CONTINUE
C
C     PACK BUFFER (B) SO STORAGE FOR AIRFOIL NAMES WILL
C     NOT LATER BE DISTURBED                                    ,48   ,49
      LEFT=IPACK(B)
C     RESERVE STORAGE FOR NAMES OF AIRFOILS                     ,50
      LAFN=IRSERV(4*NAF,B,LAFN)                                 ,51
      IF (LAFN)550,550,560                                      ,52
550   MU=3                                                      ,53
      GO TO 9000
C
C     RESERVE STORAGE FOR INTERSECTION OF LE AND TE WITH AIRFOILS ,54
560   LLETE=IRSERV(4*NAF,B,LLETE)                               ,55
      IF (LLETE)570,57C,600                                     ,56
570   MU=4                                                      ,57
      GO TO 9000
C
C     RESERVE STORAGE FOR PLANFORM POINTS, READ THEM.           ,58
600   DO 640 I2=1,2                                             ,59
      I5=2*NLTEP(I2)                                            ,60
```

```
          WING1                                                          09/14/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

      IF (IRSERV(I5,B,LPLF(I2)))610,610,620                              ,61
 610  MU=5                                                               ,62
      GO TO 9000                                                         ,63
 620  I3=LPLF(I2)                                                        ,64
      I4=I3+I5-1                                                         ,65
      READ (LI,400)(B(I),I=I3,I4)
C     LE,TE POINTS START IN B(LPLF(1)),B(LPLF(2))                        ,66   ,67   ,68   ,69   ,70
 640  WRITE (LO,650)HLETE(1,I2),HLETE(2,I2),HGIVEN,(B(I),I=I3,I4)        ,71   ,72   ,73   ,74   ,75   ,76
                                                                         ,77
 650  FORMAT(  /10HOPLANFORM A6,A3,13HEDGE POINTS ( A6,12H ARE (X,Y) =/
     .  (1H 3(2F10.4,4X)))
C
C     READ AIRFOILS AND THEIR LOCATION ON PLANFORM.
C     FIND INTERSECTION OF AIRFOILS WITH LE AND TE.
C     SCALE AIRFOILS TO FIT ON PLANFORM
 700  CALL WING1A(B,B(LAFN),LPLF,NLTEP,LOGUL,B(LLETE),LI,LO,
     .TITLE,DAT,NU)                                                      ,78
      IF(NU)710,800,710                                                  ,79
 710  MU=7                                                               ,80
      GO TO 9000
C
C     WRITE PLANFORM DETERMINED BY AF INTERSECTIONS WITH LE, TE          ,81
 800  LL=LLETE                                                           ,82
      WRITE (LO,320) DAT,TITLE                                           ,83   ,84   ,85
      DO 810 I2=1,2                                                      ,86
      L1=LL                                                              ,87
      L2=L1+4*NAF-4                                                      ,88
      WRITE (LO,650)HLETE(1,I2),HLETE(2,I2),HUSED,
     . (B(I),B(I+1),I=L1,L2,4)                                           ,89   ,90   ,91   ,92   ,93   ,94

 810  LL=LL+2
C
C     RELEASE STORAGE                                                    ,95   ,96
      DO 910 I=1,2                                                       ,97
      I3=LPLF(I)                                                         ,98
      IF (IDLETE(B(I3)))920,910,920                                      ,99
 910  CONTINUE                                                           ,100  ,101
      GO TO 1000                                                         ,102
 920  MU=8                                                               ,103
      GO TO 9000
C
C     READ PERCENTS FOR WHICH PERCENT LINES ARE TO BE GENERATED.
C     CHECK BETWEEN 0, 100.  CONVERT TO PER-UNIT VALUES.
C     STORE IN B(LPCT).                                                  ,104
1000  LPCT=IRSERV(NPL,B,LPCT)                                            ,105
      IF (LPCT)1010,1010,1050                                            ,106
1010  MU=9                                                               ,107
      GO TO 9000
C
C     READ CARD 5                                                        ,108
1050  I3=LPCT+NPL-1                                                      ,109
      READ (LI,400)(B(I),I=LPCT,I3)                                      ,110  ,111  ,112  ,113  ,114
      WRITE (LO,1060)(B(I),I=LPCT,I3)                                    ,115  ,116  ,117  ,118  ,119
1060  FORMAT(/56HOTHE FOLLOWING PERCENT CHORD LINES ARE TO BE CONSTRUCTE
```

```
             EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      .D/(1H 6F10.4))
C
      DO 1200 I=LPCT,I3                                                 ,120
      P=B(I)                                                            ,121
      IF (P*(P-100.))1200,1200,1150                                     ,122
 1150 MU=10                                                             ,123
      GO TO 9000                                                        ,124
 1200 CONTINUE                                                          ,125   ,126
      GO TO 9100
C
C     ERROR                                                             ,127
 9000 IERR=MERR(NU,NU+1,MU,WING)                                        ,128
      NU=MU
C                                                                       ,129
 9100 RETURN                                                            ,130
      END                                                               ,131
```

```
      SUBROUTINE WING1A(B,NAME,LPLF,NLTEP,LOGUL,PT,LI,LO,TITLE,DAT,MU)
C
C     USED BY WING1 TO READ AIRFOILS AND SCALE THEM TO FIT ON PLANFORM
      DIMENSION B(1),NAME(2,2,1),NLTEP(3),PT(2,2,1),RUB(2,2),LQ(2),
     1YLTE(2),NAFUL(2),CLINE(3),MX(2),HUL(2),LPLF(2),
     ,TITLE(12),DAT(2)
      LOGICAL  LOGUL(2)
C
C     THE POINTS FOR THE JTH AIRFOIL WILL BE STORED IN B(NAME(1,I,J))
C       WHERE I=1 FOR UPPER, I=2 FOR LOWER AIRFOIL.
C       THE CORRESPONDING NO. OF ELEMENTS (TWICE NO. OF POINTS) IS IN
C         NAME(2,I,J).
C     LE POINTS STORED IN B(LPLF(1)), TE IN B(LPLF(2)).
C     NLTEP(1) = NO PTS IN LE, NLTEP(2) = NO PT IN TE.
C     NLTEP(3) = NO AIRFOILS.
C     LOGUL(1) = .TRUE. IF UPPER AIRFOIL DEFINED.
C     LOGUL(2) = .TRUE. IF LOWER AIRFOIL DEFINED.
C
      DATA HUL(1)/12H UPPER LOWER/
C
 10   MU=0                                                              ,1
      NAF=NLTEP(3)
C     CLEAR NAME ARRAY                                                  ,2
      DO 100 I=1,NAF                                                    ,3    ,4
      DO 100 J=1,2                                                      ,5    ,6
      DO 100 K=1,2                                                      ,7    ,8
 100  NAME(K,J,I)=0
C                                                                       ,9    ,10   ,11   .12
      DO 8000 IA=1,NAF
C                                                                       ;13   ,14
      WRITE(LO,110) DAT,TITLE                                           ,15   ,16   ,17
 110  FORMAT(16H1WING DEFINITION 38X 2A6/1H0 12A6//)
      READ (LI,200)AFK,BETA,YLTE,AFNU,AFNL                              ,18   ,19   ,20
 200  FORMAT(6F10.0)
      WRITE (LO,210)IA,AFK,BETA,YLTE,AFNU,AFNL                          ,21   ,22   ,23
 210  FORMAT(  /12HOAIRFOIL NO. I3/6H AFK =  F4.0,12H,    AFANG = F9.4,
     X9H,    YL =F10.4,9H,     YT =F10.4/7H AFNU =F3.0,12H,     AFNL =
     X F3.0)
      NAFUL(1)=AFNU                                                     ,24
      NAFUL(2)=AFNL                                                     ,25
      IC=AFK                                                            ,26
      IF (IC*(IC-4))320,310,310                                         ,27
 310  MU=5                                                              ,28
      GO TO 9000
C
C     FIND LE AND TE X VALUES CORRESPONDING TO YLTE.
C     (ONLY 1 NEEDED IF IC=1 OR 2)                                      ,29
 320  CLINE(1)=0.                                                       ,30
      CLINE(2)=-1.                                                      ,31
      DO 350 I=1,2                                                      ,32   ,33
      CLINE(3)=YLTE(I)                                                  ,34
      ID=LPLF(I)                                                        ,35
 350  CALL POLXN(CLINE,B(ID),2,NLTEP(I),1.E-4,1,RUB(1,I),LQ,MX(I))      ,36   ,37
```

```
      IF (IC-2)400,400,500
C                                                                  ,38
 400  BETA=BETA* .01745329252                                      ,39
      SN=SIN(BETA)                                                 ,40
      CS=COS(BETA)                                                 ,41
      CLINE(1)=SN                                                  ,42
      CLINE(2)=CS                                                  ,43
      IF (MX(IC))420,420,430                                       ,44
 420  MU=6                                                         ,45
      GO TO 9000                                                   ,46
 430  CLINE(3)=-SN*RUB(1,IC)-CS*RUB(2,IC)                          ,47
      GO TO 700
C                                                                  ,48
 500  DO 530 I=1,2                                                 ,49   ,50
      IF (MX(I))520,520,530                                        ,51
 520  MU=7                                                         ,52
      GO TO 9000                                                   ,53
 530  CONTINUE
C                                                                  ,54   ,55
      CLINE(1)=RUB(2,1)-RUB(2,2)                                   ,56
      CLINE(2)=RUB(1,2)-RUB                                        ,57
      CLINE(3)=RUB*RUB(2,2)-RUB(2,1)*RUB(1,2)                      ,58
      IF (CLINE(1) .NE. 0. .OR. CLINE(2) .NE. 0.) GO TO 700
C     AIRFOIL IS A POINT                                           ,59   ,60   ,61
      DO 610 J=1,2                                                 ,62   ,63
      DO 610 I=1,2                                                 ,64   ,65
 610  PT(I,J,IA)=RUB(I,1)                                          ,66   ,67   ,68
      CHDL=0.                                                      ,69
      GO TO 820
C
C     CLINE CONTAINS COEFF. OF LINE (IN XY PLANE) WHICH
C     IS THE TRACE OF THE AIRFOIL.
C     FIND INTERSECTION WITH LE,TE AND CHORD LENGTH.               ,70
 700  DO 800 I=1,2                                                 ,71   ,72
      ID=LPLF(I)                                                   ,73
      CALL POLXN(CLINE,B(ID),2,NLTEP(I),1.E-4,1,PT(1,I,IA),LQ,MX(I))  ,74
      IF (MX(I)-1) 750,800,750                                     ,75
 750  MU=8                                                         ,76
      GO TO 9000                                                   ,77
 800  CONTINUE
C                                                                  ,78   ,79
      CHDL=DISPTA(PT(1,1,IA),PT(1,2,IA),2)
C                                                                  ,80
 820  WRITE (LO,850)((PT(I,J,IA),I=1,2),J=1,2),CHDL                ,81   ,82   ,83   ,84   ,85   ,86
                                                                   ,87   ,88   ,89
 850  FORMAT(43H LEADING AND TRAILING EDGE POINTS (X,Y) ARE/
     .2(10X,2F10.4)/15H CHORD LENGTH =F10.4)
C
C     READY TO READ, SCALE, AND STORE AIRFOILS
1000  DO 2000 IB=1,2
C                                                                  ,90   ,91
      IF (.NOT. LOGUL(IB)) GO TO 2000                              ,92   ,93   ,94
1020  N=NAFUL(IB)                                                  ,95
      IF (N)1040,1040,1100                                         ,96
```

```
          WING1A                                                          07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

 1040 IF (IA-1)1060,1060,1080                                          ,87
 1060 MU=9                                                             ,88
      GO TO 9000                                                       ,89
 1080 NB=NAME(2,IB,IA-1)                                               ,90
      GO TO 1120                                                       ,91
 1100 NB=N+N
C                                                                      ,92
 1120 NAME(2,IB,IA)=NB
C     RESERVE STORAGE FOR AIRFOIL POINTS                               ,93
      IF (IRSERV(NB,3,NAME(1,IB,IA)))1140,1140,1200                    ,94
 1140 MU=10                                                            ,95
      GO TO 9000
C                                                                      ,96
 1200 LA=NAME(1,IB,IA)                                                 ,97
      LB=NAME(1,IB,IA-1)                                               ,98
      NA=N+N
C                                                                      ,99
      CALL WING1B(LI,LO,HUL(IB),CHDL,NB,B(LB),NA,B(LA),IU)             ,100
 1250 IF (IU)1280,2000,1260                                            ,101
 1260 MU=11                                                            ,102
      GO TO 9000
C     AIRFOIL IS A POINT                                               ,103
 1280 NAME(2,IB,IA)=2                                                  ,104
      IF (CHDL)1300,2000,1300
C     SET TE POINT TO LE POINT                                         ,105
 1300 DO 1310 I=1,2                                                    ,106
 1310 PT(I,2,IA)=PT(I,1,IA)
C                                                                      ,107   ,108
 2000 CONTINUE
C                                                                      ,109   ,110
 8000 CONTINUE                                                         ,111   ,112
 9000 RETURN                                                           ,113
      END                                                              ,114
```

EXTERNAL FORMULA NUMBER - SOURCE STATEMENT - INTERNAL FORMULA NUMBER(S)

```
       SUBROUTINE WING1B(LI,LO,UL,CHORD,NB,B,NA,A,NU)
       DIMENSION B(1),A(1)
C      USED BY WING1A TO READ,WRITE,SCALE AIRFOILS
C
       IF (NA-1)400,400,200                                            ,1
 200   READ (LI,210)(A(I),I=1,NA)                                      ,2    ,3    ,4    ,5    ,6
 210   FORMAT(6F10.0)
       N=NA/2                                                          ,7
       WRITE (LO,220)N,UL                                              ,8    ,9    ,10
  220  FORMAT(  10HOTHE GIVEN I4,A6,19H AIRFOIL POINTS ARE/
      ,5X 5HXWING 17X 5HYWING)
       WRITE (LO,230)(A(I),I=1,NA)                                     ,11   ,12   ,13   ,14   ,15
  230  FORMAT(1H F11.4,F22.4)
       GO TO 600
C                                                                      ,16
 400   IF (NB-3)410,410,420                                            ,17
 410   NU=1                                                            ,18
       GO TO 9000
C      USE PREVIOUS AIRFOIL                                            ,19
 420   NA=NB                                                           ,20
       DO 430 I=1,NA                                                   ,21
 430   A(I)=B(I)                                                       ,22   ,23
       WRITE (LO,440) UL                                               ,24   ,25   ,26
 440   FORMAT(10HOTHE GIVEN A6,37H AIRFOIL IS THE SAME AS PRECEDING ONE)
C
 600   IF (NA-3)1000,1000,700
C      FIND CHORD LENGTH OF AIRFOIL                                    ,27
 700   CHDA=ABS(A(NA-1)-A)                                             ,28
       IF (CHDA)900,900,800                                            ,29
 800   IF (CHORD)900,900,1100                                          ,30
 900   NA=2
C      AIRFOIL IS A SINGLE POINT                                       ,31
 1000  NU=-1                                                           ,32
       WRITE (LO,1010)                                                 ,33   ,34
 1010  FORMAT(23HOTHE AIRFOIL IS A POINT)
       GO TO 9000
C                                                                      ,35
 1100  T=CHORD/CHDA
C                                                                      ,36
       DO 1130 I=3,NA,2                                                ,37
       IF (A(I)-A(I-2))1110,1110,1130                                  ,38
 1110  WRITE (LO,1120)                                                 ,39   ,40
 1120  FORMAT(//49HOPOINTS NOT INCREASING IN X.  JOB MAY BLOW LATER. 60X.
      1 7HWARNING///)
       GO TO 1200                                                      ,41
 1130  CONTINUE
C
C
C      SCALE A                                                         ,42   ,43
 1200  DO 1210 I=1,NA                                                  ,44
 1210  A(I)=T*A(I)                                                     ,45   ,46
       NU=0
C                                                                      ,47
```

```
          WING1B                                                            07/31/65
       EXTERNAL FORMULA NUMBER    -    SOURCE STATEMENT    -    INTERNAL FORMULA NUMBER(S)

 2000 WRITE (LO,2010)                                                    ,49   ,49
 2010 FORMAT(52HOAIRFOIL POINTS AFTER SCALING TO FIT ON PLANFORM ARE/
     ,5X 14HXWING (SCALED) 10X 1H7)
      WRITE (LO,230)(A(I),I=1,NA)
C                                                                        ,50   ,51   ,52   ,53   ,54
 9000 RETURN                                                             ,55
      END                                                                ,56
```

```
      SUBROUTINE WING2(B,LC,LGDEF,LOK,LCGUL,NAF,NPL,LAFN,KOD,PLT,
     1 NAME,PCT,CHD,EP,TITLE,DAYT,NU)
      DIMENSION NAME(2,2,1),PLT(2,2,1),PCT(1),LOK(3,2),B(1),KOD(2),NU(3)
     1,MPL(2),UL(2),TITLE(12),DAYT(2)
      LOGICAL LGDEF(3,6),LOGUL(2)
      DATA UL(1)/12H UPPER LOWER/
      DATA WING/5HWING2/
C
C     USED BY WINGA TO COMPUTE A 3D POINT ON EACH PERCENT LINE
C     AT EACH AIRFOIL AND THEN ENRICH EACH PERCENT LINE.
C
 10   NAP=3*NAF*NPL                                          ,1
      NU=0                                                   ,2
C
 50   DO 5000 KA=1,2                                         ,3
C
      IF (.NOT. LOGUL(KA)) GO TO 5000                        ,4    ,5    ,6
      LEFT=IPACK(B)                                          ,7
C     RESERVE SPACE FOR PERCENT LINE ARRAY
      IF(IRSERV(NAP,B,LOK(1,KA)))100,100,200                 ,8
 100  MU=1                                                   ,9
      GO TO 9000                                             ,10
C
 200  IA=LOK(1,KA)-3                                         ,11
C
 300  DO 4000 KB=1,NPL                                       ,12
C
      P=PCT(KB)/100.                                         ,13
      Q=1.-P                                                 ,14
C
 500  DO 3000 KD=1,NAF                                       ,15
C
      IA=IA+3                                                ,16
      LAFP=NAME(1,KA,KD)                                     ,17
      NAFP=NAME(2,KA,KD)/2                                   ,18
      MAFP=LAFP+2*NAFP-2                                     ,19
C
      IF (NAFP-1)600,700,800                                 ,20
C
 600  MU=2                                                   ,21
      GO TO 9000                                             ,22
C
 700  B(IA)=PLT(1,1,KD)                                      ,23
      B(IA+1)=PLT(2,1,KD)                                    ,24
      B(IA+2)=B(LAFP+1)                                      ,25
      GO TO 3000                                             ,26
C
 800  B(IA)=Q*PLT(1,1,KD)+P*PLT(1,2,KD)                      ,27
      B(IA+1)=Q*PLT(2,1,KD)+P*PLT(2,2,KD)                    ,28
      U=Q*B(LAFP)+P*B(MAFP)                                  ,29
      CALL BITURP(B(LAFP),B(LAFP+1),2,NAFP,KOD,U,B(IA+2),NU) ,30
      MU=3                                                   ,31
C
```

```
      EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      IF(NU)9000,3000,9000
C                                                                       ,32
 3000 CONTINUE
C                                                                       ,33    ,34
 4000 CONTINUE
C                                                                       ,35    ,36
      LEFT=IPACK(B)                                                     ,37
      KDEL=LOK(1,KA)                                                    ,38
      IF(IDLETE(B(KDEL)))4910,4920,4910                                 ,39
 4910 MU=5                                                              ,40
      GO TO 9000
C                                                                       ,41
 4920 IF(IRSERV(LEFT+NAP,B,LOK(1,KA)))4930,4930,4940                   ,42
 4930 MU=6                                                              ,43
      GO TO 9000
C                                                                       ,44
 4940 IA=LOK(1,KA)                                                      ,45
      CALL WING2P(LO,B(IA),NPL,NAF,TITLE,UL(KA),DAYT,PCT)               ,46
      CALL RICHNA(B(IA),LEFT/3,NPL,NAF,PCT,EP,CHD,KOD(2),IPL,NU)        ,47
      MPL(KA)=3*(IPL+NPL)
C                                                                       ,48
      IF(NU)4950,4960,4950                                              ,49
 4950 MU=7                                                              ,50
      GO TO 9000                                                        ,51
 4960 KDEL=LOK(1,KA)                                                    ,52
      IF(IRLEAS(B(KDEL),MPL(KA)))4970,4980,4970                         ,53
 4970 MU=8                                                              ,54
      GO TO 9000                                                        ,55
 4980 LOK(2,KA)=NPL                                                     ,56
      LOK(3,KA)=MPL(KA)                                                 ,57
      LGDEF(2,KA)= .TRUE.
C                                                                       ,58
 5000 CONTINUE
C
C     RELEASE STORAGE IN B WHICH CONTAINS THE AIRFOIL POINTS AND NAMES  ,59    ,60
      DO 8100 JA=1,NAF                                                  ,61
      DO 8100 JC=1,2                                                    ,62
      JB=NAME(1,JC,JA)                                                  ,63
 8100 JD=IDLETE(B(JB))                                                  ,64    ,65    ,66
      JD=IDLETE(B(LAFN))                                                ,67
      GO TO 9900
C                                                                       ,68
 9000 I=MERR(NU,NU+1,MU,WING)                                           ,69
      NU=MU                                                             ,70
 9900 RETURN                                                            ,71
      END                                                               ,72
```

```
          WING2P                                                                  09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE WING2P(LO,W,NPCT,NPT,TITLE,UPLO,DAYT,PCT)
      DIMENSION W(3,NPT,NPCT),TITLE(12),DAYT(2),PCT(NPCT)
C
C     USED BY WING2 TO WRITE PERCENT LINES (BEFORE ENRICHING)
C     W = ARRAY OF NPCT PERCENT LINES.  EACH LINE CONTAINS
C         NPT POINTS (X,Y,Z)
C     UPLO =6H UPPER OR 6H LOWER
C     PCT = PERCENT VALUES ASSOC. WITH EACH PERCENT LINE
C
      DATA LINPAG/45/
C
      LINREM=0
C                                                                ,1
      DO 2000 K=1,NPCT                                           ,2
      NAF=0                                                      ,3
      NP=NPT                                                     ,4
      IF(LINREM .GT. 4) GO TO 200
C                                                                ,5     ,6     ,7
  100 WRITE(LO,110) TITLE,UPLO,DAYT                              ,8     ,9     ,10
  110 FORMAT(1H1 12A6/10H POINTS ON A6,
     ,33H WING SURFACE PERCENT CHORD LINES 5X 2A6)
      LINREM=LINPAG-6
C                                                                ,11
  200 WRITE(LO,210) K,PCT(K)                                     ,12    ,13    ,14
  210 FORMAT(/24HOWING PERCENT CHORD LINE I3,1H, F9.4,8H PERCENT
     ,/7H AF NO. 7X 1HX 10X 1HY 10X 1HZ)
      N=MINO(NP,LINREM)                                          ,15
      LINREM=LINREM-N-4
C                                                                ,16
      DO 300 J=1,N                                               ,17
      NAF=NAF+1                                                  ,18
  300 WRITE(LO,310)NAF,(W(I,NAF,K),I=1,3)                        ,19    ,20    ,21    ,22    ,23    ,24
                                                                 ,25
  310 FORMAT(1H I3,4X 3F11.4)
C
      NP=NP-N                                                    ,26
      IF(NP)2000,2000,100                                        ,27
 2000 CONTINUE                                                   ,28    ,29
 9000 RETURN                                                     ,30
      END                                                        ,31
```

```
          WINGA                                                                    09/14/65
     EXTERNAL FORMULA NUMBER   -   SOURCE STATEMENT   -   INTERNAL FORMULA NUMBER(S)

      SUBROUTINE WINGA(B,LI,LO,DAT,LGDEF,LOK,LWPF,TITLE,NU)
      DIMENSION B(1),DAT(2),LOK(3,9),KOD(2),NU(3),TITLE(12)
      LOGICAL LGDEF(3,6),LOGUL(2)
C
C     USED BY GEOMD
C
 100  CALL WING1(B,LI,LO,DAT,LWPF,LOGUL,NAF,NPL,LAFN,LPCT,
     1 KOD,CHD,EP,TITLE,NU)                                               ,1
      LGDEF(1,1)=LOGUL(1)                                                 ,2
      LGDEF(1,2)=LOGUL(2)                                                 ,3
      LGDEF(1,4)= .TRUE.                                                  ,4
C
      IF(NU)1000,200,1000                                                 ,5
 200  LGDEF(2,4)= .TRUE.                                                  ,6
      LOK(2,7)=NAF                                                        ,7
      LOK(3,7)=4*NAF                                                      ,8
      LEFT=IPACK(B)
C
C     WE MUST RE-DEFINE LWPF,LAFN,LPCT.  IF THIS IS NOT DONE,THE
C     COMPILER WILL STUFF ADDRESSES AT STATEMENT 300 BASED ON
C     LOCATIONS AT STATEMENT 100.  BUT THE PACKING,
C     LEFT=IPACK(B),MAY HAVE CHANGED LOCATIONS.
C                                                                         ,9
      LWPF=LWPF                                                           ,10
      LAFN=LAFN                                                           ,11
      LPCT=LPCT
C                                                                         ,12
  300 CALL WING2(B,LO,LGDEF,LOK,LOGUL,NAF,NPL,LAFN,
     1 KOD,B(LWPF),B(LAFN),B(LPCT),CHD,EP,TITLE,DAT,NU)
C                                                                         ,13
 1000 RETURN                                                              ,14
      END                                                                 ,15
```

```
     WLDM        -  EFN    SOURCE STATEMENT  -  IFN(S)  -              05/22/67

      SUBROUTINE WLDM (NM,NROW,XP,ZP,RFAREA,AREA,XBAR,ZBAR,ALPHAM
     1,THETAM,CPM,SCL,SCD,CL,CD,CM)

C     COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT ON WING
C     SPANWISE DISTRIBUTION OF LIFT AND DRAG COEFFICIENTS

      DIMENSION AREA(1),XBAR(1),ZBAR(1),ALPHAM(1),THETAM(1),CPM(1)
     1,SCL(1),SCD(1)


      NCOL=NM/NROW
      WDRAG=0.
      WLIFT=0.
      WMOM=0.
      J=0

      DO 200 K=1,NCOL
      DRAG=0.
      XLIFT=0.

      DO 100 I=1,NROW
      J=J+1
      F=CPM(J)*AREA(J)
      XL=-F*COS(THETAM(J))                                                12
      XD=F*ALPHAM(J)
      WMOM=WMOM-XL*(XBAR(J)-XP)+XD*(ZBAR(J)-ZP)
      XLIFT=XLIFT+XL
      DRAG=DRAG+XD
  100 CONTINUE

      SCD(K)=DRAG/RFAREA
      SCL(K)=XLIFT/RFAREA
      WDRAG=WDRAG+DRAG
      WLIFT=WLIFT+XLIFT
  200 CONTINUE

      CL=WLIFT/RFAREA
      CD=WDRAG/RFAREA
      CM=WMOM/RFAREA

      RETURN
      END
```

## 6. REFERENCES


1. F. A. Woodward and J. W. Larsen, A Method of Optimizing Camber Surfaces for Wing-Body Combinations at Supersonic Speeds—Theory and Application, D6-10741, Pt. I, The Boeing Company, 1965

2. A. S. Kawaguchi, J. E. Brown, and E. LaRowe, A Method of Optimizing Camber Surfaces for Wing-Body Combinations at Supersonic Speeds—Digital Computer Program Description, D6-10741, Pt. II, The Boeing Company, 1965

3. F. A. Woodward, E. N. Tinoco, and J. W. Larsen, Analysis and Design of Supersonic Wing-Body Combinations Including Flow Properties in the Near Field—Theory and Application, D6-15044, The Boeing Company, June 1967

4. IBM 7090-7094 Programming Systems—IBJOB Processor—Overlay Feature of IBLDR, IBM File No. 7090-27—Form C28-6331, 1963